

TEBP2-YOLOv8: A Defect-Sensitive Convolutional Neural
Network Architecture Featuring Texture Enrichment and P2
Micro-Scale Detection for Fine-Grained Eggplant Quality
Assessment in Precision Agriculture

by

S. M. Ababil Islam
22301182

Debashis Saha
22101597

Md. Ridowanul Alam
24341198

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
April 2026

© 2026. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

S. M. Ababil Islam

22301182

Debashis Saha

22101597

Md. Ridowanul Alam

24341198

Approval

The thesis titled “Leveraging Convolutional Neural Networks for Eggplant Fruit Defect Detection and Classification in Agriculture” submitted by

1. S. M. Ababil Islam (22301182)
2. Debashis Saha (22101597)
3. Md. Ridowanul Alam (24341198)

of Spring, 2026 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in April, 2026.

Examining Committee:

Supervisor:
(Member)

Md. Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Bangladesh's economy is mostly dependent on agriculture. Ensuring quality control of fruits plays a crucial role in the agricultural industries of countries like Bangladesh. Eggplant is one of the most popular vegetables in Bangladesh. Detecting and classifying eggplant fruit surface defects are important for maintaining the quality of the eggplant. However, the eggplant quality assessment still relies on traditional methods, which are less accurate, time-consuming, and leads to human error. To overcome these challenges, we propose a deep learning based approach using Convolutional Neural Networks (CNNs) for the classification of eggplant fruit quality based on the condition of defects. A total of 7,686 images of different varieties of eggplants with defects from local markets and agricultural fields of Dhaka, Bogura and Satkhira districts of Bangladesh have been collected to make the dataset. This dataset has been used to train our CNN models to classify the eggplants into three categories - rotten, insect damage, and physical damage. We have initially used YOLOv8x, YOLOv8s, Faster R-CNN for training the dataset. Later, we have proposed a custom YOLOv8x model named TEBP2-YOLOv8x having an additional P2 head for improved smaller defect detection and Texture Enhanced Blocks (TEB) for texture enhancement of subtle defects. We have also used Weighted Boxes Fusion (WBF) for combining our proposed TEBP2-YOLOv8x and YOLOv8s for improved precision. Various data augmentation techniques are also used for better accuracy. After evaluating the trained models on test set images, the results demonstrate that our proposed TEBP2-YOLOv8x model has the best performance ($mAP@0.5=0.924$) among the other models. An ablation study on TEBP2-YOLOv8x model shows that it enhances the detection accuracy of insect class with tiny and subtle defects ($mAP@0.5$ increases from 0.836 to 0.904). We believe that this study can contribute to automate the eggplant defect classification with enhanced accuracy and it will reduce the economic losses in the agricultural sector by ensuring consistent eggplant fruit quality.

Keywords: TEBP2-YOLOv8, Eggplant, Deep Learning, Convolutional Neural Network, CNN, YOLOv8, YOLOv8x, YOLOv8s, Faster R-CNN, Ensemble, Weighted Boxes Fusion, WBF, Agriculture, Defect Detection, Data Augmentation, Image Classification, Texture Enrichment, Computer Vision, Object Detection, P2 Head, Multi-Scale Feature Pyramid, Texture-Enhanced Blocks, TEB, Micro-scale detection, Smaller defect Detection, Brinjal, Aubergine, Insect, Ablation Study, Precision Agriculture

Dedication

We would like to dedicate this thesis to our parents who have made our lives easier through their prayers, encouragement and sacrifices. We are who we are today because of their unwavering support.

Acknowledgement

Thanks to Almighty Allah for blessing us with patience and guidance which enabled us to complete our thesis. We want to express our gratitude to our supervisor Md. Tanzim Reza sir and co-supervisor Mr. Rafeed Rahman sir. Without their guidance, valuable advice and support during crucial times, we would not have been able to successfully complete our thesis. We want to give thanks to our honorable judging panel and BRAC University for giving us the platform and opportunity to conduct our thesis work. Finally, thanks to our parents for their prayers and support throughout our journey.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	x
List of Tables	xi
Nomenclature	xi
1 Introduction	1
1.1 Background	1
1.2 Motivation	1
1.3 Problem Statement	2
1.4 Objectives	2
1.5 Methodology in Brief	3
1.6 Scopes and Challenges	3
2 Literature Review	5
2.1 Preliminaries	5
2.2 Review of Existing Research	6
2.3 Summary of Key Findings	12
3 Requirements, Impacts and Constraints	13
3.1 Final Specifications and Requirements	13
3.1.1 Data Requirements	13
3.1.2 Annotation and Labeling Requirements	13
3.1.3 Dataset Organization	13
3.1.4 Model Performance Requirements	14
3.1.5 System Requirements	14
3.2 Societal Impact	15
3.3 Environmental Impact	15

3.4	Ethical Issues	15
3.5	Standards	16
3.6	Project Management Plan	16
3.7	Economic Analysis	17
4	Proposed Methodology	18
4.1	Design Process or Methodology Overview	18
4.2	Preliminary Design or Design (Model)	20
4.2.1	Rationale for implementing YOLOv8	20
4.2.2	Steps for YOLOv8 Performance	21
4.2.3	Architecture Design of YOLOv8	21
4.2.4	Workflow and Operations	23
4.2.5	Benefits of YOLOv8:	23
4.2.6	YOLOv8's limitations	23
4.2.7	YOLOv8 Role in Our System	24
4.2.8	Proposed TEBP2-YOLOv8x with P2 Head and Texture En- hanced Blocks (TEB)	24
4.2.9	Faster R-CNN with a ResNet-50 Backbone	26
4.2.10	WBF Ensemble (TEBP2-YOLOv8x and YOLOv8s Fusion) . .	27
4.3	Data Collection	28
4.3.1	Dataset Validation	31
4.3.2	Data Cleaning	31
4.3.3	Data Reduction and Imbalance Handling	31
4.3.4	Data Transformation	33
4.3.5	Summary of Preprocessed Data	34
4.4	Data Analysis Techniques	34
4.4.1	Accuracy	34
4.4.2	Efficiency	36
4.4.3	Reliability	36
4.5	Expected Results	37
5	Result Analysis	38
5.1	Performance of Models	38
5.1.1	Dataset Split	38
5.1.2	Results	39
5.2	Analysis of Design Solutions	50
5.2.1	YOLOv8x	50
5.2.2	YOLOv8s	53
5.2.3	Faster R-CNN	53
5.2.4	WBF Ensemble of TEBP2-YOLOv8x and YOLOv8s	54
5.2.5	Summary of strengths and weaknesses	54
5.3	Final Design Adjustments	55
5.3.1	Model-side and Data-side Adjustments	55
5.3.2	Final selection rationale	55
5.4	Statistical Analysis	56
5.4.1	Techniques applied	56
5.4.2	Relative Effectiveness Compared to YOLOv8x	56
5.5	Comparisons and Relationships	57
5.5.1	Comparison across approaches	57

5.5.2	Relationship between model capacity, defect scale, and localization	57
5.5.3	Qualitative comparison with existing works	59
5.6	Discussions	60
5.6.1	Interpretation of results	60
5.6.2	Implications for practical deployment	60
5.6.3	Limitations	60
6	Conclusion	62
6.1	Summary of Findings	62
6.2	Contributions to the Field	63
6.3	Recommendations for Future Work	63
	Bibliography	67

List of Figures

4.1	Workflow of the proposed methodology.	19
4.2	YOLOv8 Architecture [16]	22
4.3	Proposed TEBP2-YOLOv8x Architecture	25
4.4	Several captured images.	29
4.5	Several captured images.	30
4.6	Dataset Validation Certificate	31
4.7	Several labeled images	32
4.8	Initial class distribution of bounding boxes.	33
5.1	Class distribution after augmentation and class-aware oversampling.	39
5.2	Distribution of the Dataset	39
5.3	Bounding box counts across splits.	40
5.4	YOLOv8x training curves.	41
5.5	YOLOv8x confusion matrix and performance curves.	41
5.6	YOLOv8x predicted bounding boxes.	42
5.7	YOLOv8x with P2 training curves	43
5.8	YOLOv8x with P2 confusion matrix and performance curves.	43
5.9	YOLOv8x with P2 predicted bounding boxes.	44
5.10	TEBP2-YOLOv8x training curves	44
5.11	TEBP2-YOLOv8x confusion matrix and performance curves.	45
5.12	TEBP2-YOLOv8x predicted bounding boxes.	45
5.13	YOLOv8s confusion matrix and performance curves.	46
5.14	YOLOv8s predicted bounding boxes.	47
5.15	Faster R-CNN model evaluation curves	47
5.16	Faster R-CNN Predictions	48
5.17	WBF Ensemble confusion matrix	48
5.18	WBF Ensemble model evaluation curves	49
5.19	WBF Ensemble predicted bounding boxes	49
5.20	Ablation study results for mAP@50.	50
5.21	Ablation study results for mAP@50:95.	51
5.22	Ablation study results for precision.	51
5.23	Ablation study results for recall.	52
5.24	Relative Effectiveness Compared to YOLOv8x	58
5.25	Overall comparison of detection approaches.	58
5.26	mAP@0.5 comparison of the models	59

List of Tables

2.1	Comparison of Fruit Defect Detection Models	12
3.1	Project Management Plan and Timeline	16
4.1	Number of bounding boxes per defect class.	33
5.1	Class-wise bounding box counts after oversampling.	38
5.2	Final dataset split.	40
5.3	YOLOv8x baseline performance on the test set.	40
5.4	YOLOv8x with P2 performance on the test set.	42
5.5	TEBP2-YOLOv8x performance on the test set.	43
5.6	YOLOv8s performance on the test set	46
5.7	Faster R-CNN performance on the test set	46
5.8	Weighted Box Fusion (WBF) performance on the test set	50
5.9	Strengths and weaknesses of each detection approach.	54
5.10	Relative effectiveness of YOLOv8x with P2 compared to YOLOv8x baseline	56
5.11	Relative effectiveness of TEBP2-YOLOv8x compared to YOLOv8x baseline	56
5.12	Relative effectiveness of YOLOv8s compared to YOLOv8x baseline .	57
5.13	Relative effectiveness of Faster R-CNN compared to YOLOv8x baseline	57
5.14	Relative effectiveness of WBF ensemble compared to YOLOv8x baseline	57
5.15	Overall comparison of detection approaches on the test set	57
6.1	Summary of mAP@0.5 across evaluated models	63

Chapter 1

Introduction

1.1 Background

Fruit grading is an important part of harvesting, but manual grading is time-consuming and requires human labour [7]. In the present world, improving productivity in agriculture has become one of the crucial challenges [17]. Fast and accurate detection is important to increase efficiency and reduce food waste. Eggplant is one of the most popular and widely grown vegetables in Bangladesh. Many studies focus on eggplant diseases during the harvesting stage. However, there is only limited research on eggplant defect detection after they are collected from the field. There is also not sufficient defect detection study on local Bangladeshi Eggplant image dataset. The detection of smaller and subtle defects is also challenging. So, our study focuses on eggplant defect detection and classification using deep learning-based Convolutional Neural Networks on locally collected dataset and solving the smaller defect detection limitation.

1.2 Motivation

Eggplant (*Solanum melongena*) comes from the Solanaceae family and the *Solanum* genus. It provides essential vitamins, minerals, and dietary fiber, which are crucial for our human health and well-being [4]. Eggplant is one of the most common and popular vegetables in Bangladesh. But eggplants are very delicate, and insects can damage them at any stage. Also, eggplants can easily get scratches during picking, packing, or transport. These kinds of defects reduce the quality and the market value of the fruits. Bangladeshi farmers, buyers, and sellers check eggplants manually with their eyes and hands. This method is slow, inefficient, and not always correct. Moreover, detecting tiny and subtle defects like insect holes and scratches, is very difficult to identify for the human eye. This not only affects the quality assessment and profit but also increases food waste. Therefore, we need a solution to find defective eggplant including smaller and subtle defects in a faster, accurate, and more reliable way, which will help to minimize the food waste as well as help

the farmer to maximize their profit. A deep learning-based model is helpful in this case. It has robust data processing capabilities [15]. Convolutional Neural Network (CNN) is a popular deep learning based algorithm that is widely used in computer vision related tasks like object detection, image recognition [10]. CNN-based object recognition models, such as R-CNN, YOLO, SSD can extract and learn complex image patterns from the raw images [3]. Therefore, using CNN-based models in this case is the ideal way to extract features, learn patterns, and identify eggplant defects. This CNN-based eggplant defect detection model will be an easy and reliable way to detect and classify eggplant defects.

1.3 Problem Statement

Traditional image processing methods require manual feature design. It uses features like edge detection, color analysis to identify objects. However, Convolutional Neural Networks (CNNs) methods do not require manual feature design. Instead, it learns features automatically from the training images[14]. Hence, deep learning-based Convolutional Neural Network (CNN) methods are better than the traditional image processing methods. It gives more accuracy and reliability. Our model is likely to provide better and accurate results compared to the normal image processing models. Previous research may detect fruit and vegetable defects and freshness in different environmental conditions. However, those environmental conditions are likely not the same as in Bangladesh. Therefore, it will not provide the optimal results. In our study, we collect images from the local fields and markets. It may give more optimal results than the previous research models. In Bangladesh, farmers use their hands and traditional equipment to collect eggplants from the field. Therefore, eggplant physical damage defects vary in both appearance and shape. The insect damages are also tiny and subtle which are challenging to detect. To address this issue, we develop a custom model focusing on enhancing smaller and subtle defect detection accuracy. We collect a wide range of eggplant images with various kinds of defects. This helps the model give more accurate results in real life.

1.4 Objectives

The aim of this research is to improve eggplant quality control assessment using deep learning. Our main goal is to develop a deep learning-based model using Convolutional Neural Networks (CNNs) for eggplant defect detection and classification. Main objectives of our research –

- The model will classify eggplant fruits into three different labels. These labels are rot, insect and physical. It will help to easily identify healthy and defective eggplants.

- We will collect and label a vast number of eggplant images having different types of defects from local fields and markets. It will help the model to detect defects accurately under the local conditions. Also, the images are captured with different lights, angles, and defect types so the model works well in real situations.
- We will propose a custom CNN-based model to enhance the detection accuracy of smaller and subtle defects such as tiny insect and physical damages.
- We will evaluate and compare the proposed model’s performance with other baseline models using precision, recall, mAP and conduct ablation study on our proposed model to identify its effectiveness.

1.5 Methodology in Brief

Our research followed an explorative, computer vision-based approach to detect eggplant defects. For this purpose, we selected deep learning-based methods. We used local markets and bazaars for data collection. We collected healthy eggplants and eggplants with various defects, such as rot, physical damage, and insect damage, which ensured data diversity and the efficiency of the result. we tried to take several classes of data and many images of data. After that, we preprocessed our collected data. We resized eggplant images to ensure consistent pixel dimensions. We fixed the pixel values and applied data augmentation to improve the model’s performance. For model selection, we chose YOLOv8 and Faster R-CNN deep learning models. In this study, we proposed a custom YOLOv8x architecture with a P2 head and TEB block to improve defect detection accuracy and named it TEBP2-YOLOv8x. We conducted ablation study to show the effectiveness of our model. After that, we ensemble our TEBP2-YOLOv8x model with YOLOv8s using weighted box fusion. We adjusted the loss function, epoch values, and other hyperparameters during the training process. We used Python as the main programming language and libraries like PyTorch. After training the models, we evaluated and compared them using metrics such as mean average precision, precision, and recall. We collected the data ourselves, which led to some issues with the background and lighting. We resolved data imbalance using oversampling and data augmentation.

1.6 Scopes and Challenges

Scopes

- This research is only limited to eggplant defect detection. It does not include other types of fruits or vegetables.
- The model uses the external appearance of eggplants. It does not cover any internal problems or chemical issues.

- Dataset is collected from the local markets and agricultural fields in Bangladesh. This model may not work properly or give an accurate result in a new environment.

Challenges

- Collecting a large set of eggplant images from various angles, lighting conditions and defect types may be difficult.
- Some defects are too small or not clearly visible, which makes the detection task more difficult.
- Making our proposed model not only easy to use but also budget friendly is a big challenge.

Chapter 2

Literature Review

2.1 Preliminaries

Defect Classification

Fruit quality is commonly evaluated in agricultural postharvest system depending on both internal and exterior criteria. External flaws are the first markers of the quality of the fruits that's why these are very important. Defect is any surface level irregularity including physical damages like cracks, bruising, rotten or fungal infected, insect damage with tiny holes etc which can be caused by biotic or abiotic reasons. In our study, Eggplant fruit defects are classified based on these causes into three categories such as rot, physical and insect.

Deep Learning and CNNs

Deep learning is a sub branch of machine learning that improved the image processing system from dataset produced by images collected directly from fields by enabling end to end feature learning. Convolutional Neural Networks or CNNs are the deep learning models that are effective for detecting textures and spatial hierarchies in raw images. CNNs have layers including convolutional layers, pooling layers, and fully connected layers which use techniques like backpropagation, large-scale training data to learn features like patterns of the defects, color gradients, edges. These networks automatically learn abstract features, such as edges, color gradients, and defect patterns.

Object Detection Models

For detecting eggplant defects, different kinds of object detection models can be used. CNN models including YOLO, R-CNN, Faster R-CNN, VGGNet, ResNet, DenseNet, MobileNet are some well known models for object detection which provide

different level of accuracy, depth and efficiency. Models such as YOLO, Faster R-CNN, and SSD are commonly known models for real time object detection. These models help the system to predict bounding boxes of the objects which enable fast detection of different classes of defects in eggplant or other fruits.

Evaluation Metrics

The performance of the models can be measured using different types of evaluation metrics such as,

- **Precision:** It indicates the model's proportion of the predicted positives that are true positive.
- **Recall:** It means the proportion of actual positives.
- **F1 score:** Harmonic mean value of precision and recall is known as F1 score.
- **Mean Average Precision (mAP):** It's a summary metric which is mainly used for tasks related to object detection.

2.2 Review of Existing Research

Several recent studies have been conducted that directly involved in defect and disease detection of eggplant using deep learning methods. Wang et al.[32] presented an unique approach to detect disease of eggplant fruit using multimodal data fusion with attention mechanism which enhanced accuracy. They collected image data with using environmental sensor inputs like light, CO₂, humidity and temperature for thoroughly representing the features. They combined both online images and real photos from fields to produce a dataset with total of 6,658 images including five eggplant diseases. For balancing the dataset, applied data augmentation like CutMix, Mosaic, and GridMask has been utilized with sensor data. The authors used ResNet-50 CNN model for the image feature extraction and BERT model for processing the sensor data. An attention mechanism model has also been integrated for features related to eggplant disease that reduced irrelevant features and a weighted fusion method has been introduced for combining the features. During the training phase, they also used an embedding loss function for aligning image and sensor features more efficiently. This model achieved 94% precision, 90% recall and 92% accuracy. This result surpassed the performance of the models like DETR and YOLOv8, especially when there was a critical scenario.

In this paper [11] Saad et al. proposed an automated approach for the recognition of eggplant disease which utilized transfer learning based on convolutional neural network. They collected a dataset consisting of 2,766 images which were labeled and categorized into 16 classes, among them 14 categories are disease affected and 2 are healthy categories. For improving the model generalization, various data augmentation techniques have been used including zooming, rescaling, shearing, flipping

and rotation. The authors trained the system using three pre-trained models like DenseNet201, Xception, and ResNet152V2. Among the three models, DenseNet201 performed better than the remaining models. This model has accuracy 99.06%, precision 99.45%, specificity 99.49% and sensitivity 87.30%. During spider mite and leaf roller identification, it performed with accuracy of 100%. The model split the dataset with 80% train data and 20% test data. It trained the data with softmax activation, cross-entropy loss and Adam optimizer which has learning rate 0.001. This paper represents a more comprehensive and reliable technique than many other previous works that conducted using smaller datasets and less classes. The authors used metrics like false positive rate (FPR), false negative rate (FNR) and multiclass confusion matrices to evaluate the performance of the model. The evaluation results clearly indicate that DenseNet201 can be very accurate for high quality classification performance because of its feature reuse and dense connectivity.

The paper authored by Liu and Wang [30] proposed a deep learning based system named EggplantDet for real time eggplant disease detection. They used YOLOv8 framework for this system. Their EggplantDet consists of the FasterNet backbone for enhancing the computational process. For better sensitivity when detecting the minor disease features, they integrated a 160×160 object detection head. The researchers also utilized Triplet Attention Mechanism (TAM) and WIoU loss function which improved the model's feature extraction and bounding box regression respectively. They collected a dataset with 4,930 labeled images which were categorized into total 5 classes for training and testing the model. After training, the system EggplantDet provided result with mAP 92.61%. The detection speed of the system achieved 88.39 FPS. This result of proposed EggplantDet exceeded baseline YOLOv8s by mAP 6.89% with FPS 2.91. It also kept the parameter count only 10.99M. By performing other experiments, the system also proved that it can outperform other SOTA models like R-CNN, YOLOv5, YOLOX. By achieving 90% accuracy, their model also generalized better when testing with other datasets consisting of tomato, cucumber. The impact of each module of the system was positive for detecting small objects when evaluated by ablation studies.

Rangarajan et al. [5] introduced an enhanced classification method based on VGG16 deep learning model that can classify five eggplant diseases especially in leaf area including Tobacco Mosaic Virus (TMV), Cercospora leaf spot (CLS), Epilachna beetle (EB), Two-Spotted Spider Mite (TSSM) and Little Leaf (LL). A collection of 2,815 field images was used for producing the dataset. Various augmentation techniques like brightness, rotation, translation, zooming, cropping were used to increase the original images to 8,852 which was used for training the model. The model optimized to achieve validation accuracy 99.4% and test accuracy 94.3%. The initial learning rate was 0.0001 using 16 batches and 10 epochs. A Multi-class Support Vector Machine (MSVM) has been used for extracting and evaluating the feature parameter of higher layers of VGG16 model. It clearly showed 25 to 31 layers were enough to classify CLS, TMV, LL AND TSSM having accuracy 100%. But in case of EB, the performance was inconsistent with highest performance of 97.7%. They also found 50 channels which can be more dominating than others especially channel 419 whose accuracy 81.42%. From these particular channels, almost 10000 parameters set which have reduced features have an accuracy of 99.3%. The authors also deployed the model to smartphone after optimizing to Tensorflow Lite.

The research paper [8] written by Haque and Sohel mentioned another model that used two-stream fusion architecture with CNN-SVM and CNN-Softmax for eggplant disease detection. The study conducted in Bangladesh and the dataset collected from local fields as well as online sources consisting RGB images of 2,284. Their dataset have been produced that included 9 eggplant disease such as Phomopsis blight, *Cercospora melongenae*, aphids, tobacco mosaic virus etc. The produced dataset was augmented using some techniques like scaling, rotation, flipping and cropping to increase its number to 9,620. The authors used Transfer Feature Learning method for implementing Transfer Learning that utilized a total of 6 pretrained CNN models. The models included MobileNet, ResNet50, V3, VGG16, VGG19 and NasNetMobile. The layers of the model trained and optimized using the new fully connected layers and for solving the issue of overfitting, they used an approach of adaptive learning rate which based on loss trends. After training and testing, the result showed in their paper that Inception V3 has the highest accuracy of 96.11%. The other top two model were VGG19 and MobileNet having accuracy of 93.74% and 92.17% respectively. However, the accuracy of these top three models enhanced to 98.9% when they implemented the interface model to them. Their study also indicated other evaluation metrics like in case of F1 score with the fusion model the Inception V3 performed more than 97% when it was tobacco mosaic virus and aphids and 81% for *Cercospora melongenae*. For Inception V3 and MobileNet, t-SNE method used for better feature separability. Though their proposed fusion model used more computational time than usual but it outperformed well enough when compared to individual CNNs.

After exploring research papers from multiple sources, we noticed that the study for defect detection of eggplant fruits using deep learning is very limited, but many other fruits like Apple, Tomato, Guava etc. have been used by the researchers widely for defect or disease detection. Nithya et al. [9] authored the paper that introduced an computer vision system for defect detection of mango surface defects using deep learning models. They utilized a dataset which was available publicly consisting images of 100 Kent mangoes. Among them 50 were healthy and remaining 50 were defective. For reducing noise and improving contrast, Adaptive Wiener filtering and histogram equalization methods have been implemented for data preprocessing. Data augmentation methods such as flipping, rotation, scaling were also utilized to increase the image dataset to 800 images which solved the issue of insufficient data and made the dataset more effective for training the model. Having 6 convolutional, 5 max pooling layers and 2 fully connected layers, there were total of 13 layers that have been used for making the custom CNN architecture for their automated system. They also used softmax and ReLU activation that ensures binary classification and various size kernels like 3×3 , 4×4 , and 8×8 for extracting feature maps. They preserved features with high values when degradation of the samples done using max pooling method. For achieving the most efficient and optimal accuracy from the model, they trained the model with training rate of 0.001 and optimized with Adam optimizer. The proposed model by the authors showed result that has 98.5% mean accuracy, 98.5% specificity, 98.25% sensitivity and 0.98 AUC value. The evaluation have done using cross validation method which had 10-fold and all folds had no more than 3 misclassification. Some other classical method like SVMs, fuzzy incremental learning, the accuracy ranged between 87% to 97.88%, but the

proposed method had performed significantly well comparing with these methods. They proposed this system that can replace manual feature extraction with fully automated one and assured that this type of system can be suitable for other fruits also.

Wang and Xiao [6] developed an system of lychee surface defect detection that was implemented by deep convolutional neural network (DCNNs) and the system integrated by a data augmentation model named generative adversarial network (GAN). Total of 3,743 labeled lychee images have been used for making a strong dataset which was classifies into three categories such as mature, defects and rot consisting 44.03%, 25.75% and 30.22% respectively of the dataset. Transfer based GAN which can synthesize images, was used later in the system for solving class imbalance by augmenting the classes into 1400 images. They trained total of three object detection models named Faster R-CNN with ResNet-50, SSD-MobileNet V2, Faster R-CNN with Inception-ResNet V2. After evaluating the model, it showed results for all the models. Among them SSD-MobileNet V2 achieved mean average precision of 91.81% with interface speed of 102 FPS. These result of the model indicated that the model is appropriate for industrial purposes for real time detection of defects. After training the model with post augmentation datas, they also evaluated the performance of each models. In case of mAP of SSD-MobileNet V2 increased by 2.86%, Faster R-CNN with Inception-ResNet V2 increased by 0.58% and Faster R-CNN with ResNet50 increased by 1%. Implementing GAN-based augmentation technique also noticeably decreased detection gaps of inter class. For example, maP gap of SSD-MobileNet V2 decreased from 9.45% to 1.78%. Activation heatmaps and other qualitative results also indicates that this GAN augmentation based technique enhanced object localization. Though using more complex Faster R-CNN models showed better accuracy than the GAN based model, but the detection speed is lower as the they showed 23 FPS speed while GAN based model showed speed of 68 FPS. The study indicated that the SSD-MobileNet V2 can be the best solution for real time detection of Lychee surface defect detection as it can balance both the accuracy and efficiency

Chiou et al. [27] proposed another system for defect detection of guava fruit in their research paper that is based on deep learning models. They collected 189 guavas of *Psidium guajava* L. species from fields and markets and they took total of 1,701 images of the guavas with resolution of 3840×2160 pixels that labeled and categorozied into 13 classes with different kinds of defect such as sunburn, pest damage, fungal infections etc. YOLOv4-COCO pretrained convolutional neural network has been used. Total of 1,190 images of the dataset have been utilized for training purpose and 511 images for testing purpose. System with Intel i7 CPU and RTX 3080 GPU were used for training the model. It took about 50 hours to complete the training. After performing the evaluation, the model provides accuracy 88.15%, recall 94.36%, precision 92.44% and F1 score 93.39%. But the model's performance fluctuated when it was class specified. It showed 0% to 38% false negative rates and 0% to 36% false positive rates. The error of the system increased for classes that were not represented during the training. When using 1000×1000 focused region, the system detected defect of 13×14 size which was smallest with speed of 12 FPS. The model demonstrated efficiency when identifying biological features. For instance, It provided false positive and negative rates less than 9% when detecting

fungal classes such as *Pestalotiopsis psidii*, *C. gloeosporoides*, *Phyllosticta psidiicola*. The model also showed strong generalization when it tested with other 200 images which were irrelevant from their specified classes.

The paper [20] from Patidar and Chakravorty highlighted a novel lightweight Convolutional Neural Network with four layer (Conv-4 DNCC) based disease detection system of apple fruits. This system can detect three types of diseases that found in apple fruit such as rot, blotch and scab. Their dataset comprised of 5,915 images which was collected from Kaggle and local fruit markets. To enhance the diversity of the dataset they used augmentation techniques like shifting, zooming, shearing and flipping. The system's architecture built upon three types of layers such as convolutional, pooling and fully connected. They also integrated ReLU activations and dropout layers. These added layers helped enhancing regularization. These layers trained by Adam optimizer with more than 20 epochs which had learning rate of 0.0001 and 56 batch size. It also trained by categorical cross entropy loss. The result after evaluating the model demonstrated that it achieved 95.37% accuracy, 95.37% precision, 95.31% recall and 95.25% F1 score. The model showed minimal error in classification after confusion matrix analysis. The introduced architecture of the model optimized computational overhead. For this reason, this system is highly suitable for deploying in devices with low resource.

Pacheco et al. [13] used original and synthetically generated images as datasets for their paper to demonstrate comparison between automatic defect detection of apples and mangoes based on convolutional neural network models. They collected approximately 20,000 images in which 10,000 were real and remaining 10,000 were generated synthetically. This dataset were used for training and evaluating the models like InceptionV3, DenseNet121, MobileNetV2 and VGG16. They specified four classes including scabs for apples, black spots for mangoes, rotten and bruises. Before training the models they preprocessed the datas by standardizing and filtering manually. They also augmented the images by rotating and adjusting the contrast. The models trained over 36 sessions for both apples and mangoes data. During the training, they used epochs of 10, 20 or 30. They split the dataset into 80%, 10%, 01% for training, validating and testing respectively. MobileNetV2 provided 97.5% accuracy for apples and 92.9% accuracy for mangoes which was best among all the models. From the other models, InceptionV3 and DenseNet121 achieved more than 90% accuracy in case of both of the fruits. But VGG16 showed poor performance for defect detection of mangoes having 63.6% accuracy. When 30 epochs with drop out rate 0.2 have been used, the models gave optimal training output. The study further demonstrated that CNNs which are lightweight like MobileNetV2 are highly suitable for binary classification of defects and they also computationally efficient models.

Knot et al. [12] proposed a model using machine learning which will detect fruit quality using image processing. The data used are image data. There were some limitations for this paper because of insufficient data and also they got accuracy of 95.8% with the help of Convolutional Neural Networks which is considered high accuracy. For this model, it needed a high and large labeled dataset and GPU intensive training. That is why the author added an alternative which is Vision Transformers(ViTs) which works as a fixed feature model. These things are classified

as SVM, Logistic regression and XGBoost. They used two dataset, one is banana ripeness dataset having almost 273 data and 4 classes. Another one is the apple dataset . For this they took almost 5858 image data and they maintain class as healthy vs worm-damaged. They used models such as ResNet-101, DenseNet, ViT based pipeline. Specifically Dino vIT S/8 connected with XGBoost which provided almost 94% accuracy for the banana dataset. Also for the apple they achieved 95% accuracy and they used ViT based model. There they saw that the ViT model required sample training three times fewer to achieve 90% . Also CNN does outperform on low-data. Therefore, PCA showed that Dino ViT gives a good performance under those data which are noisy and also inconsistent. In this model pipeline's don't need retraining of vit model and also need low weight classifiers. Which is very helpful and can be used in standard hardware in agriculture systems. So it can be very important in the means of affordable perspective. It can remove the high-end infrastructure and large labeled dataset .

In another paper [25] authored by Azadpour et al. proposed a method of grading oleaster fruits using deep learning based framework. They tried to introduce a better method for grading instead of traditional mechanical methods. They made the dataset containing 2000 oleaster images. The dataset was divided into four classes following national standard of Iran. MASK R-CNN algorithm was implemented during segmentation phase. This phase achieved instance segmentation accuracy (ISA) oscillating from 99.86% to 112.40% and object detection rate (ODR) of 100%. For classifying quality of the fruit two types of YOLOv8 models such as YOLOv8n and YOLOv8x have been utilized when training the model with the segmented data. Among two of the models, YOLOv8x had showed 97.30% sensitivity and 91.25% validation accuracy. But YOLOv8n model performed better showing 100% sensitivity and 98.33% specificity. YOLOv8n also achieved 92% accuracy and changing sensitivities between 87.10% to 94.89% when performing in real-time. Each sample had mean total processing time of 877.98ms. The architecture of Mask R-CNN integrated with YOLOv8n successfully showed real-time classification without reducing accuracy and had 105 kg/h operational throughput making it highly suitable for deployment in industrial sectors for better fruit grading.

Kujur et al. [29] addressed an hybrid citrus fruit disease detection system based on deep neural networks. Their framework integrated Convolutional Neural Networks (CNNs) for its capability of feature extraction and Gradient Boosting (GB) for its high performance. The authors divided their collected dataset having 3000 images of citrus fruits into healthy, Canker, Blackspot, Greening disease. The images of the dataset resized and augmented using data augmentation techniques for robustness. ResNet50 CNN model was used in their custom architecture for the purpose of feature extraction. This architecture used to make flattened feature vectors from the dataset images which were classified by implementing XGBoost algorithm. The authors evaluated the model using four optimizers such as Adam, Adamax, RAdam and Nadam. Among these four optimizers, Nadam performed better than other optimizers in case of almost all matrices. The combination of CNN, GB and Nadam optimizer showed 98.03% accuracy, 98.02% recall, 98.04% precision and 98.00% F1 score. Categorical cross entropy as loss function, ReLU activation and Softmax have also been used in this model. The other combinations of the model like CNN and SVM or CNN and RF underperformed than combination of CNN, GB and

Nadam optimizer. Even this framework performed superior than approaches based on VGG16 which had 93.66% accuracy. This study validated CNN, GB and Nadam based model as the most efficient solution for detection of citrus diseases among the other tested combinations of the models.

2.3 Summary of Key Findings

The reviewed literature indicates that deep learning models especially CNNs can be implemented efficiently for defect detection of fruits. CNN architectures such as MobileNet, ResNet, VGGNet, DenseNet are widely used in different researches for this kind of task which shows high accuracy in classification tasks between 90% to 98%. These features can efficiently identify surface level different types of features and defects. The studies mostly utilized RGB images which captured under prebuilt environment. Some preprocessing techniques like resizing, zooming, changing contrast, data augmentation etc. are used to enhance the performance of the models. However, there were also some limitations that can be noticed such as high performance hardware limitation for large dataset and also the collecting images from field with different lighting and environment is challenging. Most of the studies used fruits such as apples, mangoes, guavas that indicates that there is research gap for classification system with eggplant dataset. In summary, different CNNs can be highly efficient for our defect detection and classification of eggplant based dataset which implementation can be cost effective and beneficial in agriculture.

Table 2.1: Comparison of Fruit Defect Detection Models

Fruit Disease Detection Models: Dataset and Performance Comparison					
Fruit	Author (Year)	Model / Architecture	Dataset Size	Classes	Accuracy / Performance
Eggplant	Wang et al. (2025)	ResNet-50 + BERT (Multimodal Fusion)	6,658	5 diseases	92% Acc., 94% Prec.
	Saad et al. (2022)	DenseNet-201 (Transfer Learning)	7,266	16 (14D + 2H)	94.5% Precision
	Liu & Wang (2025)	CNN + YOLOv8 + Fast R-CNN	4,930	5 classes	98.2% Accuracy
	Rangarajan (2021)	VGG16 + MSVM	8,852	5 diseases	99.49% Accuracy
	Haque & Sohil (2022)	Two-stream Fusion CNN	–	9 diseases	98.9% Accuracy
Mango	Nithya et al. (2022)	Hybrid CNN + Attention	–	Binary	98.5% Accuracy
Lychee	Wang & Xiao (2021)	SSD-MobileNet V2 + GAN	3,743	3 classes	95.8% mAP
Guava	Chiou et al. (2025)	YOLOv4 (COCO Pretrained)	1,701	13 defects	88.15% Acc., 93.39% F1
Apple	Patidar (2024)	CNN (AlexNet, VGG, ResNet, Inception)	9,915	4 classes	95.37% Accuracy
	Pacheco et al. (2023)	MobileNetV2 (Transfer Learning)	20,000	–	97.5% Accuracy
	Knot et al. (2023)	ViT + XGBoost	~6,100	Healthy / Diseased	94% Accuracy
Watermelon	Azadpour (2025)	CNN + YOLOv8	2,000	4 grades	95.4% Accuracy
Citrus	Kupur et al. (2025)	Hybrid CNN-ResNet50 + XGBoost	3,000	4 diseases	98.03% Accuracy

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Our study focuses on developing an automated eggplant defect detection and classification system using deep learning-based models such as YOLOv8 and Faster R-CNN. We set several technical and functional requirements for this research in order to ensure reliable, consistent, and practical outcomes.

3.1.1 Data Requirements

We have collected 7,686 images from local markets and agricultural fields to make our final dataset. Blurry, dark, and unclear images are removed manually from our dataset to maintain dataset quality. We sort eggplants into three categories based on the defect types. These three classes are: rot, insect and physical.

3.1.2 Annotation and Labeling Requirements

We use Roboflow to annotate and label our dataset. All eggplant images are manually annotated using bounding boxes. We label each bounding box according to its defect type. We review bounding box labels to reduce annotation errors and ensure consistency throughout the annotation process.

3.1.3 Dataset Organization

The eggplant dataset is divided into three subsets: training, validation, and testing. Models are first trained using the training set. Then hyperparameters are optimized,

and the final performance is evaluated using validation and test sets, respectively. We also monitor each class distribution to identify class imbalances. Data augmentation techniques such as rotation, flipping, and scaling are used to reduce this class imbalance and improve model generalization.

3.1.4 Model Performance Requirements

We expect our model to detect and classify eggplant defects in complex real-life conditions. Standard performance metrics such as precision, recall, and mean Average Precision (mAP) are used to evaluate model performance. We evaluate the performance of the proposed TEBP2-YOLOv8x, YOLOv8s and Faster R-CNN models separately. We conduct ablation study on TEBP2-YOLOv8x. Then, we ensemble the custom YOLOv8x and YOLOv8s models to improve predictive performance. Our goal is to maintain a balance between performance and computational complexity.

3.1.5 System Requirements

Hardware Requirements

- AMD Ryzen 9 5950X CPU (16-core, 32-thread)
- NVIDIA RTX 3090 GPU (24 GB VRAM)
- 64 GB DDR4 RAM
- 1 TB SSD storage for model training and development
- Three smartphone cameras (Realme GT Master Edition, Samsung F23, Infinix Hot 50) for image capture

Software Requirements

- Programming language: Python 3.8
- Deep learning framework: PyTorch
- Object detection models: Ultralytics YOLOv8, Faster R-CNN
- Image processing libraries: OpenCV, Python Imaging Library (PIL)
- Dataset annotation: Roboflow
- Version control: GitHub

3.2 Societal Impact

This automated eggplant defect detection and classification study can bring significant benefits to Bangladesh’s agricultural sector. Farmers can identify eggplant defects early, which helps to reduce post-harvest losses. They can get higher market prices and increase their income. Market vendors and retailers can sort eggplants more quickly and efficiently. They can maintain consistent quality standards and reduce customer complaints. It also helps them reduce labor costs. Rotten, physically damaged, and insect-damaged eggplants are removed before sale. Thus, customers get only healthy eggplants. Furthermore, Rotten and insect-damaged vegetables spread various foodborne illnesses. This study helps to lower the risk of foodborne illness and supports national food security.

3.3 Environmental Impact

This study has various positive environmental effects. It helps to reduce waste and encourages farmers to use more sustainable farming practices. In traditional eggplant quality assessment, a large portion of eggplants is discarded due to late or inaccurate defect detection. Our proposed study can help to identify defects before transportation, storage, or market distribution. As farmers can filter out spoiled and defective eggplants just after harvesting, it significantly minimizes transportation costs. It also indirectly reduces fuel usage and carbon dioxide emissions. An automated defect-detecting method allows farmers to store only healthy eggplants in cold storage. This saves electrical energy and reduces costs. Additionally, this study helps to encourage environment friendly farming practices. Farmers can make the right decisions by analyzing past defect statistics of the eggplants. It helps them to use pesticides efficiently and lower chemical usage. It not only protects water and soil but also preserves natural enemies and biodiversity. Overall, automated eggplant defect detection and classification improve resource utilization and support long-term environmental conservation.

3.4 Ethical Issues

Ethical considerations play an important role in this research. All images were collected responsibly from local markets and fields without violating privacy. No personal or private data were collected. Our dataset does not contain any personal, confidential, or sensitive information. This research aims to support farmers and vendors rather than entirely reduce human labor. We collected image data from multiple districts and under varying lighting conditions to ensure fairness and reduce bias. The model provides clear bounding box outputs around defective areas. This reduces the blind reliance on automated decisions. In conclusion, the proposed defect detection model ensures privacy, fairness, and ethical use of artificial intelligence.

3.5 Standards

Several widely accepted standards are followed in this study to ensure quality and reliability. The dataset is annotated in standard YOLO and COCO format. We maintain transparency and fairness throughout the research. To ensure reproducibility, we record learning rate, number of epochs, batch size, and optimizer choice. The performance of the model is evaluated using standard performance metrics such as precision, recall and mAP.

3.6 Project Management Plan

Name	Description	Duration(2025 - 2026)
Related Work Study	Literature review on related papers, identify problem statement, methodology analysis.	February - April
Dataset Collection and annotation(Phase - 1)	Collect 3000+ images from local markets and Kawran Bazar; bounding box annotation and metadata standardization.	May - June
Model Training (YOLOv8)	Train the YOLOv8x and YOLOv8s baseline model and evaluate detection performance.	July - August
Dataset Collection and annotation(Phase - 2)	Collect 5500+ images from Dhaka, Bogura, and Satkhira; image labeling and fix previous annotations.	August - October
Model Training (Faster R-CNN)	Train the Faster R-CNN model and evaluate performance.	November - December
TEBP2-YOLOv8x Model Development	Dataset size and annotation adjustment, develop TEBP2-YOLOv8x model and conduct ablation study on it, train other baseline models	January - March
Model Ensemble (WBF)	Implement Weighted Boxes Fusion (WBF) ensemble of TEBP2-YOLOv8x and YOLOv8s; evaluate and compare final performance.	March - April

Table 3.1: Project Management Plan and Timeline

3.7 Economic Analysis

The proposed eggplant defect detection and classification system offers economic benefits to the agricultural supply chains. Although the initial cost of this study may seem high, it is beneficial in the long run. The data collection, model training, and computing hardware costs are mostly one-time expenses. However, these costs are significantly lower than the overall manual sorting process and the losses caused by defects. Therefore, in the long run, this study will benefit farmers, sellers, customers, and the agricultural sector as a whole.

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

The goal of this research's design process is to use convolutional neural networks (CNNs) to create an automated system for detecting and classifying eggplant fruit defects. There are few steps on methodology and the main part of the methodology is dataset collection, data preprocessing, designing model, training, evaluation and the last part is result analysis.

We carefully prepared all of the steps such as accuracy, find out efficiency and the reliability of the classification process. Initially, we collected a large number of healthy and defected eggplants from the local market and after that we collected photos by using mobile phones. The raw image was almost 8000+ and later, the number of images was reduced to 7,686 which was used as the final dataset. To maintain diversity, we maintain a few tricks and those are various lighting sources, orientation of eggplant, surface characteristics, considered every part of eggplant. Each image was labeled by Roboflow which is a very renowned platform for data augmentation and labeling. Three different classes: rot, insect and physical damage were manually applied to each image. The healthy images are considered as null when labeling with roboflow. We maintained the labeling accuracy and labeled quality by taking instructions from our supervisor . After that we used our dataset which was labelled by Roboflow. When our dataset was ready we performed some preprocessing steps which made our dataset more uniform and also made the image better. We did preprocessing by resizing, normalizing, and reducing noise from image to make the model more robust and less likely to overfit, data augmentation methods like rotation, flipping, and changing the brightness were used. It helped the overall model more efficiently and learned to know some data which had not been seen before. We initially used standard YOLOv8x model to find and identify defects because they are the best balance of speed and accuracy. We also proposed TEBP2-YOLOv8x model implementing P2 head for small defect detection and a Texture Enhanced Block (TEB) for improving sensitivity of surface defect textures. We conducted ablation study on our proposed TEBP2-YOLOv8x model to evaluate performance of each proposed components. We also used YOLOv8s and Faster

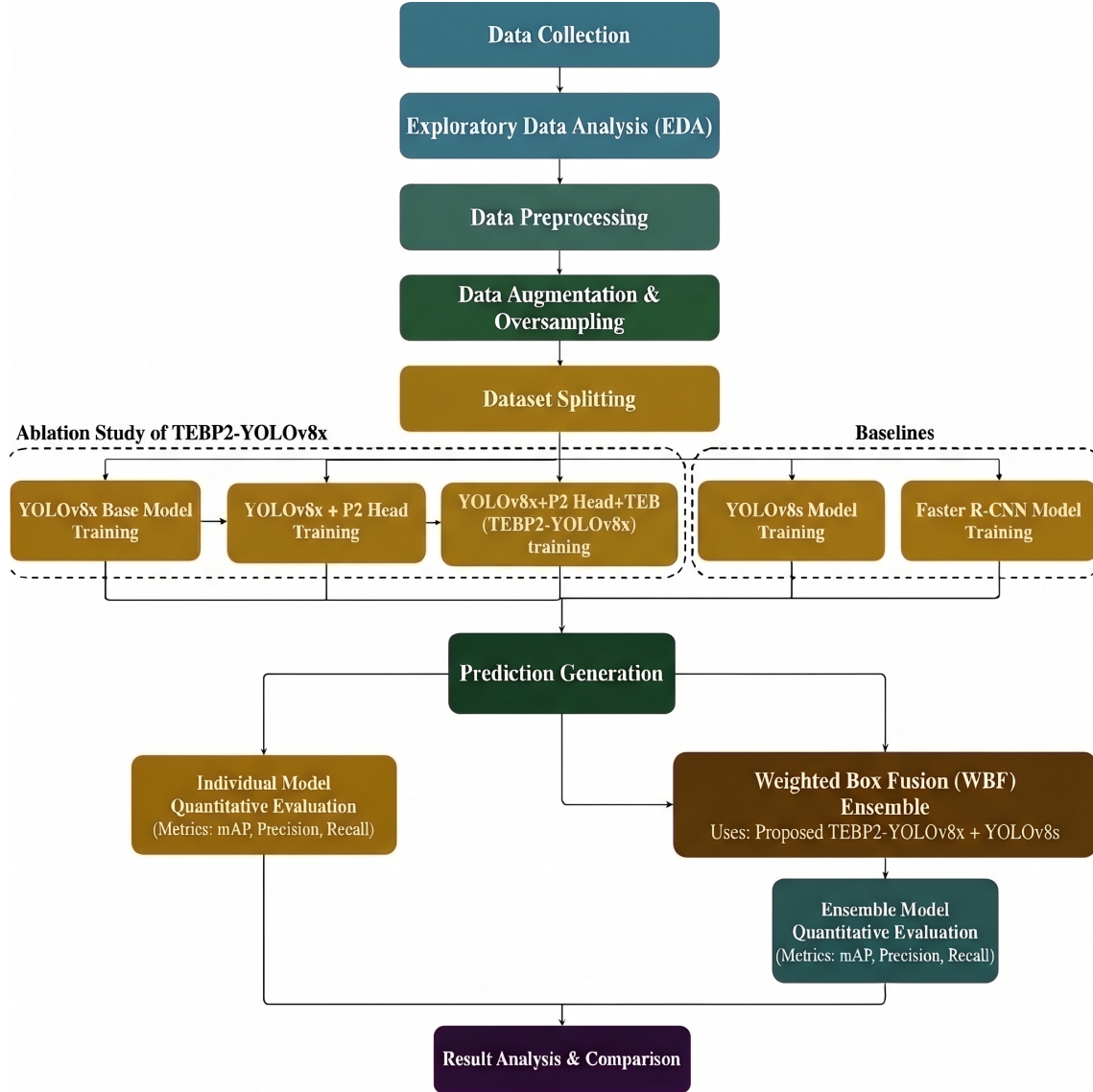


Figure 4.1: Workflow of the proposed methodology.

R-CNN as baseline models to compare with the proposed model. The YOLOv8s model we used is very efficient and doesn't need much computing power and it is much faster [18]. We implemented Weighted Boxes Fusion or WBF ensemble using our TEBP2-YOLOv8x and YOLOv8s to improve the precision. After the model were trained, they were tested using standard performance measures like precision, recall and mean Average Precision (mAP). We also looked at the other output like graphs. Finally, the trained model were tested with real-world eggplant images to see if they worked well in real-world. This methodological framework shows how attached CNN-based architectures with the YOLOv8 detection system can quickly result in the process of finding defects in eggplants. The method helps improve the overall outcome of fruit quality, it also minimizes the risk which can be caused by humans, and enhances agricultural productivity in Bangladesh through advanced AI-based control of quality.

4.2 Preliminary Design or Design (Model)

Model architectures were evaluated at and tested in order to make the Deep Learning-Based Eggplant Fruit Defect Detection and Classification System. A modular design method was suggested that uses deep learning and image processing to find and sort defects. This part talks about the overall workflow such as our models, which algorithm we used and also the framework which helped us to develop the model. One of the most important steps in the whole process is analyzing and finding eggplant defects in locating surface defects like rot, insect damage, and physical damage. To achieve this, we chose object detection which is the main method because it can find and sort multiple defect areas in an image at the same time. We chose the YOLOv8 model for this project because it is very fast and accurate after looking at some other different architectures. To find the right balance between detection accuracy and computational efficiency, we used YOLOv8x and proposed TEBP2-YOLOv8x model. Roboflow was used to label the dataset, and eggplant images were put into four groups: healthy (labeled as null or background), rotten, insect-damaged, and physically damaged. This design makes sure that eggplant defects are categorized properly which helps make the quality control process in farming automatic and reliable.

4.2.1 Rationale for implementing YOLOv8

We chose the YOLOv8 model for detecting and classifying defects in eggplant fruit because it is very accurate, efficient, and works well with real-world agricultural datasets. YOLOv8 can accurately find multiple defect areas in one image by drawing bounding boxes around each area. Precise localization is important for finding small or overlapping defects, which are common in eggplants. YOLOv8 preserves the spatial and contextual information by analyzing the entire image as opposed to individual regions [24]. When defects are partially hidden or visually appear to healthy areas, this helps the model better understand the connection between the various parts of the fruit. Because of natural variations, occlusions, and changing

lighting, agricultural datasets frequently have few images or noisy samples. Reliable detection and classification are ensured by YOLOv8's resilience in situations like this, which allows it to learn significant features even from small or flawed datasets. YOLOv8 outperforms many other cutting-edge object detection models with a fairly low computational resource demand. Faster training and inference are made achievable by this balance, which makes it possible to use genuine settings with even mid-range GPUs or high-end devices.

4.2.2 Steps for YOLOv8 Performance

The following methodical procedures are used by YOLOv8 to identify and categorize defects:

- **Grid Division:** A grid of cells is generated from the input image, and each cell is responsible for identifying defects in its assigned region.
- **Bounding Box Prediction:** For defects that are detected, each grid cell makes a prediction about the size, coordinates, and confidence levels of potential bounding boxes.
- **Class Probability Estimation:** Probabilities for the identified class healthy, rotten, insect or physically damaged are assigned to each predicted box.
- **Non-Maximum Suppression :** Redundant predictions for the same defect are eliminated by filtering overlapping bounding boxes and maintaining only the ones with the highest confidence scores.
- **Final Classification:** The image's detection and classification output is complete when the remaining bounding boxes are finalized with the appropriate defect categories. [22]

4.2.3 Architecture Design of YOLOv8

The advanced object detection model YOLOv8 is made for identifying and categorizing objects in real time. It is ideal to detect eggplant surface flaws in a single forward pass because it combines convolutional neural networks (CNNs) with anchor-free detection and current optimizations. For real-world agricultural applications, YOLOv8 strikes a balance between speed, accuracy, and computational efficiency.

Core Architecture components:

1. **Feature extraction (backbone):** YOLOv8 extracts hierarchical features from the input image using CSP (Cross Stage Partial) networks. Both fine-grained details like insect damage and larger patterns like rot or physical defects are captured by multiple convolutional layers with different kernel sizes (usually 3x3 and 1x1) [23]. In order to maintain gradient flow and avoid vanishing gradients, residual connections are applied.

4.2.4 Workflow and Operations

1. **Preprocessing the input:** The input images are normalized and resized to 640 x 640 x 3. To improve generalization, data augmentation methods like flipping, rotation, and brightness adjustment are used.
2. **Backbone Feature Extraction:** Convolutional layers based on CSP process input. Low-level characteristics of eggplants, like their edges, shapes, and color variations, are preserved by early layers. High-level features like shapes, patterns, and defect structures are captured by deeper layers.
3. **Fusion of Features (Neck):** PANet is used to combine multi-scale features for improved context preservation.
4. **Head for Detection:** Bounding boxes, confidence scores, and class probabilities are predicted for every grid cell. The most reliable detections remain when overlapping boxes are removed using Non-Maximum Suppression (NMS).
5. **Classification:** The four defect classifications are used to categorize final bounding boxes. For every class, a sigmoid activation is used to calculate the class probabilities.
6. **Training and Loss Functions:** YOLOv8 implements a variety of loss functions: Bounding box loss for orientation (CIoU or GIoU). Loss of objectiveness to identify flaws. Classification loss for correctly classifying defects.[21]

4.2.5 Benefits of YOLOv8:

- **Precise Localization:** Precise bounding boxes for multiple flaws in one image.
- **Context Preservation:** Identifies errors without losing track of small objects or spatial relationships.
- **Efficiency on Noisy and Small Datasets:** Maintains a strong performance even when handling noisy or small datasets.
- **Less computationally demanding:** Moderate GPUs can detect in real time.
- **Single-Pass Detection:** For speed, this method combines classification and localization into a single forward pass.

4.2.6 YOLOv8's limitations

YOLOv8 needs enough labeled data in order to train with high accuracy. It may overlook very minor or hidden flaws if augmentation is not sufficient. In specialized datasets, anchor-free design may require careful modification to specific object sizes [31].

4.2.7 YOLOv8 Role in Our System

The main model for identifying and categorizing eggplant defects in this study is YOLOv8. The model consistently produces outputs for automated quality assessment by accurately recognizing and localizing a variety of defect types. Its design guarantees precise detection of both minor and major flaws, allowing for useful implementation in agricultural environments.

4.2.8 Proposed TEBP2-YOLOv8x with P2 Head and Texture Enhanced Blocks (TEB)

YOLOv8x with its standard architecture has some limitations when detecting very small surface defects such as the tiny insect holes or small physical or rotten area of our eggplant images. These defects cover smaller regions that can be lost during deeper feature extraction stages. To address these limitations, we customized the model and introduced two main modifications such as: adding of P2 detection head which have stride 4 and integration of Texture Enhanced Blocks (TEB). These modifications are proposed to enhance the accuracy of detecting small defects and texture patterns.

Integration of P2 Detection Head

The standard YOLOv8x model has three feature scales for detection which are P3 with stride 8, P4 with stride 16 and P5 with stride 32. These scales are not suitable for detecting tiny defects. For improving the detection of smaller defects we introduced an additional high resolution feature map P2 having stride 4.

Modified Feature Pyramid Network (FPN)

For integration of P2 level, the top-down feature pyramid is extended and the up-sampled P3 feature map is combined with the corresponding P2 feature map from the backbone. Because of this, the model can preserve fine spatial details during extraction of semantic information from deeper layers.

Extended PAN Path

The path aggregation network (PAN) is also extended for effective feature propagation. The P2 feature map is downsampled adding it with the higher feature maps which ensured information flow with all the scales.

Detection Head Expansion

Finally, the detection head is expanded to four feature scales such as P2, P3, P4 and P5 having stride number respectively 4, 8, 16 and 32.

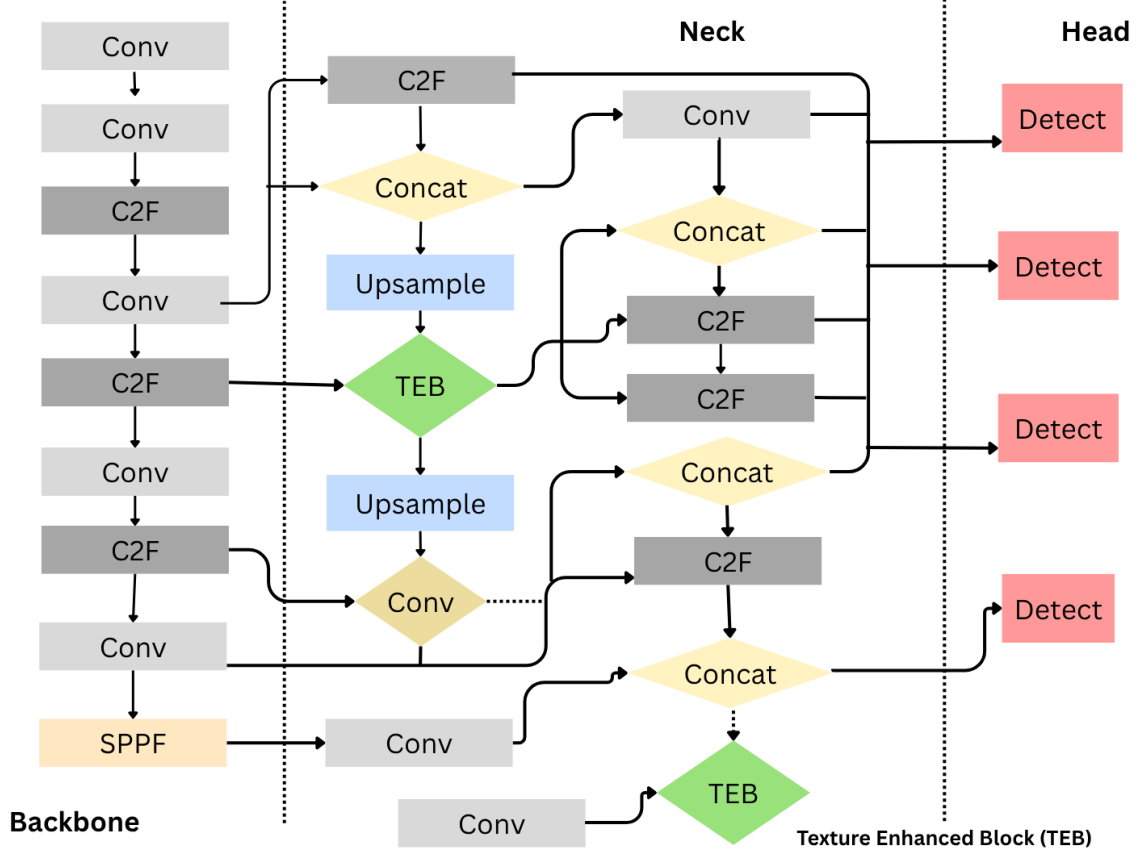


Figure 4.3: Proposed TEBP2-YOLOv8x Architecture

Texture Enhancement Block (TEB)

The Texture Enhancement Block or TEB is introduced for further enhancing the subtle defects detection. This module is a lightweight module to improving detection of fine grained spatial features such as textures and local intensity variations. We introduced the TEB module with three main stages:

- 1. Depthwise and Pointwise Convolution:** Depthwise convolution is firstly used for processing input feature maps which is followed by pointwise convolution, batch normalization, and activation. Localized spatial patterns are detected with efficient computation in this stage.
- 2. Channel Attention Mechanism:** Adaptive average pooling is used in TEB module to extract global information from feature map. Then the informations passed through a lightweight transformation for creating channel-wise weights. This allows our model emphasizing important feature channels.
- 3. Residual Feature Enhancement:** Through a residual connection, enhanced features and the original input are combined. This preserves important information while selectively enhancing subtle texture features.

The TEB modules are included with the P2 and P3 feature levels because of their higher spatial resolution which is more suitable for detecting fine grained textures. TEB was not added on deeper layers as those layers are responsible for extracting semantic information rather than textures.

Ablation Study Design for Proposed TEBP2-YOLOv8x model

An ablation study is necessary to conduct for evaluating and comparing the effectiveness of each proposed component of the TEBP2-YOLOv8x model. For this purpose, three model variants are designed such as:

- **A0:** Baseline YOLOv8x which is the standard YOLOv8x model.
- **A1:** YOLOv8x with P2 detection head.
- **A2:** YOLOv8x with P2 and TEB module which is our final proposed TEBP2-YOLOv8x model.

This ablation study is useful for understanding the contribution of proposed P2 head for small defect detection and TEB for texture enhancement.

4.2.9 Faster R-CNN with a ResNet-50 Backbone

We also used Faster R-CNN model with a ResNet-50 Backbone. YOLOv8 model is faster and slightly less accurate. On the other hand, Faster R-CNN is relatively slower, however it has high accuracy. Faster R-CNN has a more complex architecture than YOLOv8. It allows considering all the edge cases while making decisions. It is developed to address the limitations of previous models such as R-CNN.

Faster R-CNN working process:

- **Region Proposal Network (RPN):** It identifies potential object regions by using sliding window.
- **Classification and Bounding Box Regression:** It uses the features from ResNet-50 to classify each region and adjust bounding boxes.
- **End-to-End Training:** The RPN and classifier are trained together. It makes the model more efficient.

Advantages:

- It is suitable for complex and diverse images, such as agricultural images.
- It has high accuracy due to the ResNet-50 layers.

Disadvantages:

- It has slower inference speeds compared to one-stage models such as YOLOv8.
- It requires more computing power.

4.2.10 WBF Ensemble (TEBP2-YOLOv8x and YOLOv8s Fusion)

The Weighted Box Fusion (WBF) ensemble technique combines predictions from multiple models to improve object detection results. In this study, the WBF ensemble fuses predictions from our proposed TEBP2-YOLOv8x and YOLOv8s. YOLOv8x model is best for accuracy [19]. YOLOv8s is faster as it is lightweight [26]. The system may provide better results by using the WBF ensemble.

Process:

- Prediction Merging: In our study, our TEBP2-YOLOv8x and YOLOv8s provide labeled bounding boxes as outputs. WBF considers the intersection of two bounding boxes. It combines overlapping boxes into a single prediction.
- Box Confidence: Each prediction has a confidence score. WBF assigns more weight to predictions with a high confidence score.

Benefits of the WBF Ensemble:

- Improved mAP (Mean Average Precision): The fusion helps to improve Mean Average Precision. It also helps to achieve a balance between false positives and false negatives.
- Robust Performance in Complex Scenarios: It improves the overall performance, particularly in some edge cases where defects are hard to predict for a single model.

Drawbacks of the WBF Ensemble:

- Increased Computational Cost: Running multiple models and then combining their results requires more computational power.
- Low-End Edge Devices: WBF ensemble requires heavy computational power. Therefore, it is unsuitable for low-end devices with limited processing capabilities.

4.3 Data Collection

Our data collection process involved multiple steps. First, we analyzed the local eggplant market and identified major types of eggplant defects. For the data collection, our main goal was to create a dataset that can perform well in the local environment. There were many free datasets available online for eggplant defects. But those eggplant varieties were not our local varieties. They are not suitable for our research as we wanted to use the dataset that will represent our local eggplant varieties. So, we made the entire eggplant image dataset from the local Bangladeshi markets to train the model. This helped the model to work accurately and reliably in our environment. We captured eggplant images from Dhaka, Bogura and Satkhira districts of Bangladesh to create a diverse dataset. We have primarily collected images of three different local varieties of eggplants such as Uttara, Nayantara, ISD 006. Therefore, images vary in size, shape, color, and surface. We captured images under natural lighting environments using three smartphone cameras: Realme GT Master Edition, Samsung F23, and Infinix Hot 50. Each camera has different specifications. We captured each eggplant image from various angles and distances. These steps helped to create a balanced dataset. It also improved the model's ability to perform under different lighting and imaging conditions. We divided eggplants into three defect classes based on the visible condition and considered the images without defect as healthy. The three defect classes are:

Physical: Eggplants that have visible cuts and scratches.

Rot: Eggplants that show dark spots and fungus.

Insects: Eggplants that have bite spots and holes.



Figure 4.4: Several captured images.



Figure 4.5: Several captured images.

4.3.1 Dataset Validation

The prepared image dataset was validated and reviewed by Dr. Md. Rafiqul Islam, Professor and Chairman, Department of Plant Pathology, Sher-e-Bangla Agricultural University. He ensured that the annotation and the bounding boxes of the defects of the eggplant images were correct and consistent.

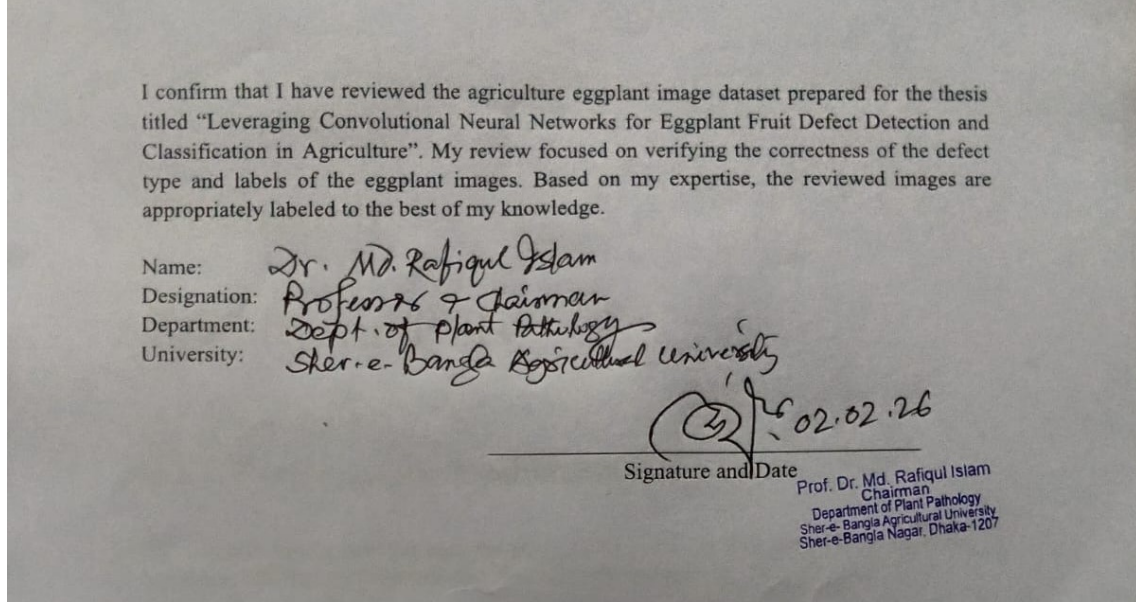


Figure 4.6: Dataset Validation Certificate

4.3.2 Data Cleaning

We checked each image to maintain quality and consistency. Since low-quality images do not provide any useful information for the model. Therefore, we first removed all blurry images from our dataset. We also removed duplicate images from the dataset since they can create bias. We excluded poor lighting conditions, such as harsh sunlight or dark shadow images, from the dataset. Since we collected and labeled all the images manually using Roboflow, no missing values were present in the dataset. After the cleaning process, only clear and high-quality images were present for further preprocessing and the model training process.

4.3.3 Data Reduction and Imbalance Handling

The healthy images were represented as null or background. We had to remove some of the healthy images to balance the dataset. After implementing a class frequency analysis on our collected dataset, we noticed that the insect and rot classes were underrepresented, while the physical samples were overrepresented. We fixed this issue by using an imbalance handling method. We also used class-aware oversampling, in which minority classes were replicated and enhanced more than the majority classes.



Figure 4.7: Several labeled images

This process made sure that every class made an equal contribution to the model optimization process [2].

Initial class distribution of bounding boxes:

Class	Bounding Boxes
Insect Damage	3,615
Physical Damage	5,753
Rotten	2,941
Total	12,309

Table 4.1: Number of bounding boxes per defect class.

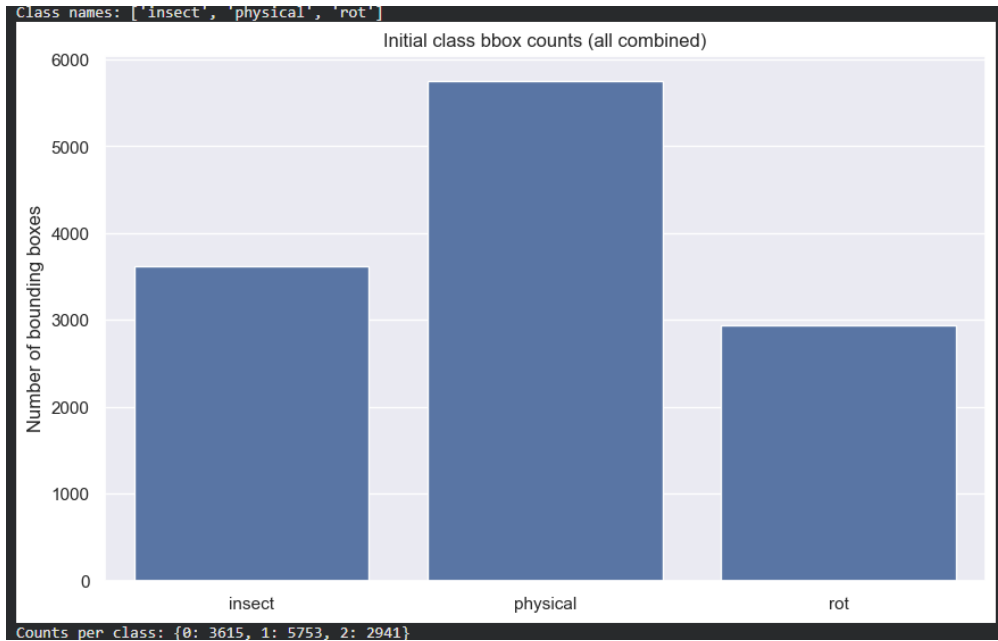


Figure 4.8: Initial class distribution of bounding boxes.

4.3.4 Data Transformation

The preprocessing stage standardizes image appearance to reduce variance caused by lighting and capture conditions. In our pipeline we apply:

- Normalization
- Contrast enhancement
- Gamma correction to reduce lighting variation
- Geometric and photometric augmentation to increase diversity

We performed data transformation to prepare the dataset for training. The main goal was to make the dataset consistent and increase diversity. We scaled all image

pixel values between 0 and 1. Gamma Correction and Contrast Limited Adaptive Histogram Equalization (CLAHE) were applied to improve image contrast and details. We applied data augmentation on the dataset to increase diversity and reduce overfitting. In the data augmentation process, we applied horizontal flipping, rotation, scaling, shifting, blurring, and Gaussian noise. Lastly, we encoded the class label in YOLO format and saved augmented images and labels in separate folders for training, validation, and testing.

4.3.5 Summary of Preprocessed Data

A refined and balanced dataset was produced after implementation of data preprocessing and handling the imbalance. The dataset consists of images with four categories: insect, rot, physical damaged and healthy which are considered as background or null when labeling. The main characteristics of our dataset after all the preprocessing are:

- All the duplicate, blurry and inconsistent labels have been removed from the dataset and reduced the spatial dimension of the images into 640 x 640
- Every image has been subjected to gamma correction, pixel intensity normalization and contrast enhancement using CLAHE.
- Images from overrepresented class were removed and class imbalance solved using class-aware oversampling.
- Images were augmented through various transformations like rotation, shifting, Gaussian noise, blurring.

4.4 Data Analysis Techniques

To evaluate model performance we particularly focus on four dimensions such as accuracy, efficiency, reliability and testing methodology.

4.4.1 Accuracy

Detection accuracy in this work is measured using standard object detection metrics based on *Intersection over Union (IoU)* matching between predicted bounding boxes and ground-truth boxes. For a predicted box B_p and a ground-truth box B_{gt} , IoU is defined as:

$$\text{IoU}(B_p, B_{gt}) = \frac{\text{Area}(B_p \cap B_{gt})}{\text{Area}(B_p \cup B_{gt})}, \quad (4.1)$$

where $\cap$ denotes the overlap region and $\cup$ denotes the union region.

IoU-based Matching and Confusion Counts

If a predicted box matches a ground-truth box of the same class with $\text{IoU} \geq \tau$ for a fixed IoU threshold τ , it is regarded as a correct detection (True Positive, TP). To prevent double counting, each ground-truth box can be matched to *at most one* prediction (and vice versa).

Unmatched ground-truth boxes are counted as False Negatives (FN), and unmatched predictions are counted as False Positives (FP). Standard detection assessment pipelines are supported by this IoU-based matching protocol. [1]

Precision and Recall

Given TP, FP, and FN at threshold τ , we compute Precision and Recall as:

$$P(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau)}, \quad R(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)}. \quad (4.2)$$

Precision penalizes false detections (FP), while Recall penalizes missed objects (FN). [1]

Average Precision (AP)

Average Precision (AP) summarizes the PR curve into a single scalar by measuring the area under (a possibly interpolated) PR curve. A widely used classical definition (e.g., PASCAL VOC 2007) uses the 11-point interpolation:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p_{\text{interp}}(r), \quad (4.3)$$

where the interpolated precision is:

$$p_{\text{interp}}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}). \quad (4.4)$$

This interpolation smooths small oscillations in the measured PR curve and yields a stable AP estimate. [1]

Mean Average Precision (mAP)

Mean Average Precision averages AP across C classes:

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c. \quad (4.5)$$

mAP@0.5 : $mAP@0.5$ denotes mAP when the matching threshold is fixed to $\tau = 0.5$:

$$mAP@0.5 = \frac{1}{C} \sum_{c=1}^C AP_c(\tau = 0.5). \quad (4.6)$$

mAP@0.5:0.95 : $mAP@0.5:0.95$ denotes the average of mAP over a set of IoU thresholds $\mathcal{T} \subseteq [0.5, 0.95]$:

$$mAP@0.5:0.95 = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \left(\frac{1}{C} \sum_{c=1}^C AP_c(\tau) \right). \quad (4.7)$$

Higher IoU thresholds require closer box alignment with the ground reality, which complicates this metric. [28]

Metrics Reported,

Based on the above definitions, we report: (i) Precision (P), (ii) Recall (R), (iii) $mAP@0.5$, and (iv) $mAP@0.5:0.95$, applying IoU-based matching between ground truth and predictions in accordance with accepted detection requirements.[1], [28]

4.4.2 Efficiency

Efficiency shows how well a model meets throughput requirements for applications in real life such as sorting and grading. We evaluate efficiency through:

- Inference latency (time per image),
- Computational demand (heavier models require stronger GPUs/CPUs),
- Pipeline complexity (single-model vs ensemble; post-processing costs).

Our TEBP2-YOLOv8x and YOLOv8s provide measurable inference time per-image (in milliseconds). Because of its two-stage architecture, faster R-CNN usually has higher inference costs, and the WBF ensemble adds overhead because it needs to run multiple detectors in place of fusion.

4.4.3 Reliability

Consistency across defect types and operational situations is referred to as reliability. Reliability in agricultural photos is impeded by:

- Illumination variation (sunlight, shadows, indoor lighting)

- Reflections on glossy eggplant skin
- Variation of background clutter
- Variety in defects (particularly small insect damage)

We use confusion matrices, qualitative visual evaluation, and per-class performance to assess reliability.

4.5 Expected Results

The goal of this study is to achieve strong performance while maintaining computational efficiency. Our expected outcomes are:

Enhanced Tiny Insect Defect Detection Accuracy with Proposed TEBP2-YOLOv8x: The YOLOv8x model can accurately localize and classify any type of defect. It is expected that our proposed TEBP2-YOLOv8x can also achieve enhanced precision and recall. YOLOv8 model is more powerful in detecting larger defects, such as rot. We expect that our TEBP2-YOLOv8x having P2 head and TEB can also detect smaller defects like insect tiny holes more accurately. Overall, this model is expected to create a balance between accuracy and computational complexity.

Improved Performance with WBF Ensemble: The WBF Ensemble combines predictions of the TEBP2-YOLOv8x and YOLOv8s models. It is expected to improve detection Precision. The ensemble method will boost overall accuracy, but it may also bring computational complexities.

Efficient Real-Time Performance: The YOLOv8s model is a smaller and faster version of YOLOv8x. Due to its smaller architecture, it has less accuracy. However, it can still perform well when real-time performance is crucial. An automated sorting system requires accuracy with high speed. Therefore, it can be used in automated sorting system.

Environmental Variability: We captured images from multiple districts to make our dataset. We applied various preprocessing techniques and data augmentation methods. It allows the models perform well under different lighting conditions and environments. However, extreme lighting conditions or heavy reflections can disrupt defect detection.

Insect Defect Detection: Small defects, such as insect damage, are expected to be challenging for all models to detect. Small defects have subtle visual patterns. It is harder to capture, label, and detect. Therefore, insect damage is expected to be slightly less accurate than other defect types.

Chapter 5

Result Analysis

5.1 Performance of Models

5.1.1 Dataset Split

The overall pipeline for dataset preparation consists of preprocessing, augmentation, and class-aware oversampling to address imbalance.

- 10897 total images after augmentation.
- 4779 oversampled images created for balancing.

Class	Bounding boxes
Insect	11498
Physical	11605
Rot	10143

Table 5.1: Class-wise bounding box counts after oversampling.

The dataset has been stratified and split into three subset train, validation and test keeping the ratio 75:15:10.

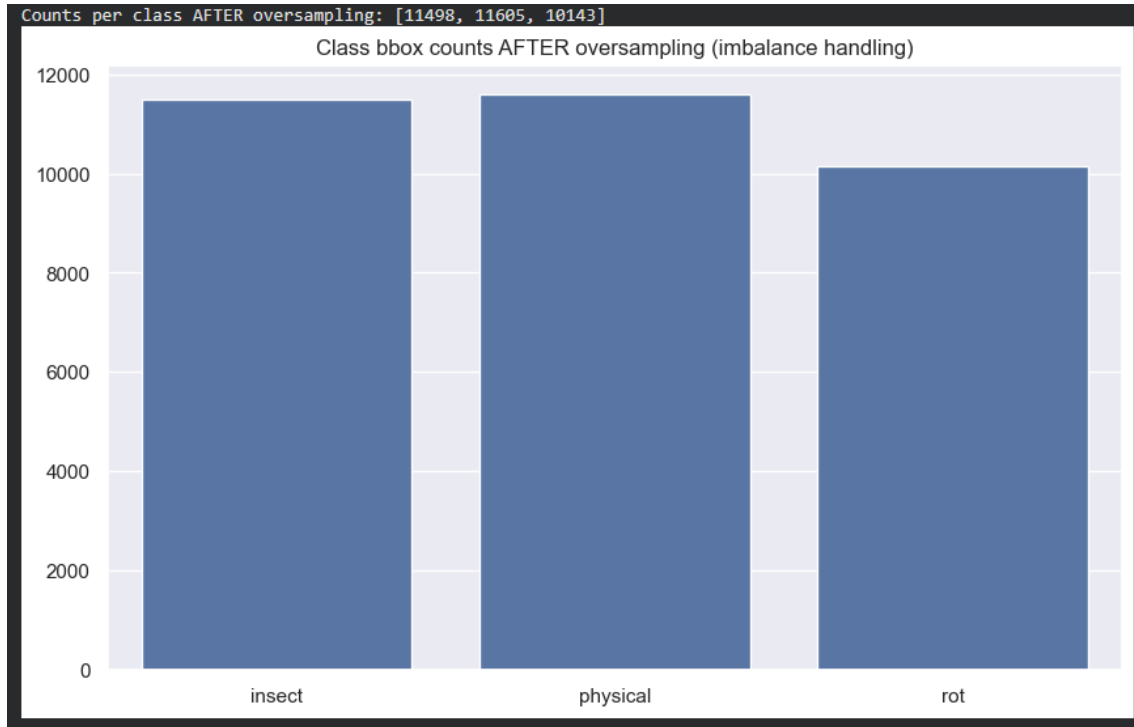


Figure 5.1: Class distribution after augmentation and class-aware oversampling.

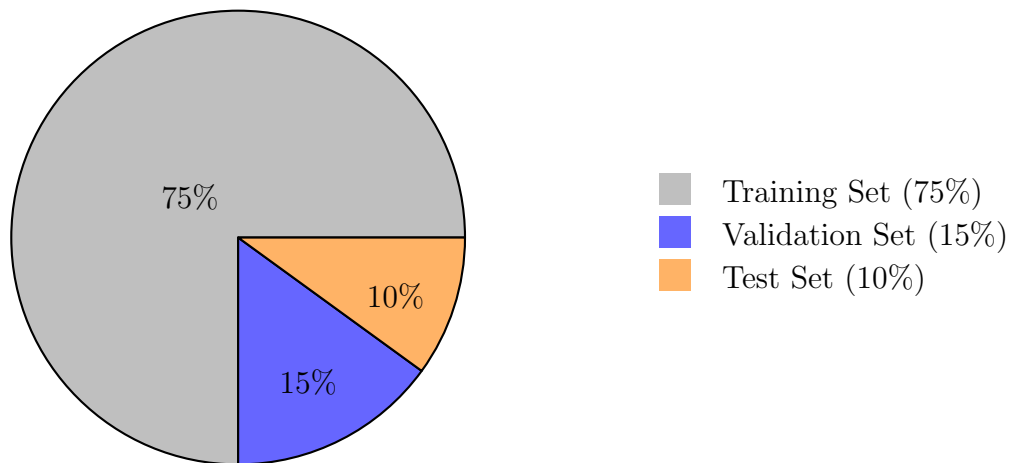


Figure 5.2: Distribution of the Dataset

The final dataset contains 15676 images and is split into: Train = 11756; Validation = 2352; Test = 1568.

5.1.2 Results

YOLOv8x performance on the test set

YOLOv8x performs best on rot, likely because rot regions usually have stronger contrast changes and the defect size is comparatively larger. However, Insect damage is challenging because it is small and has subtle texture changes.

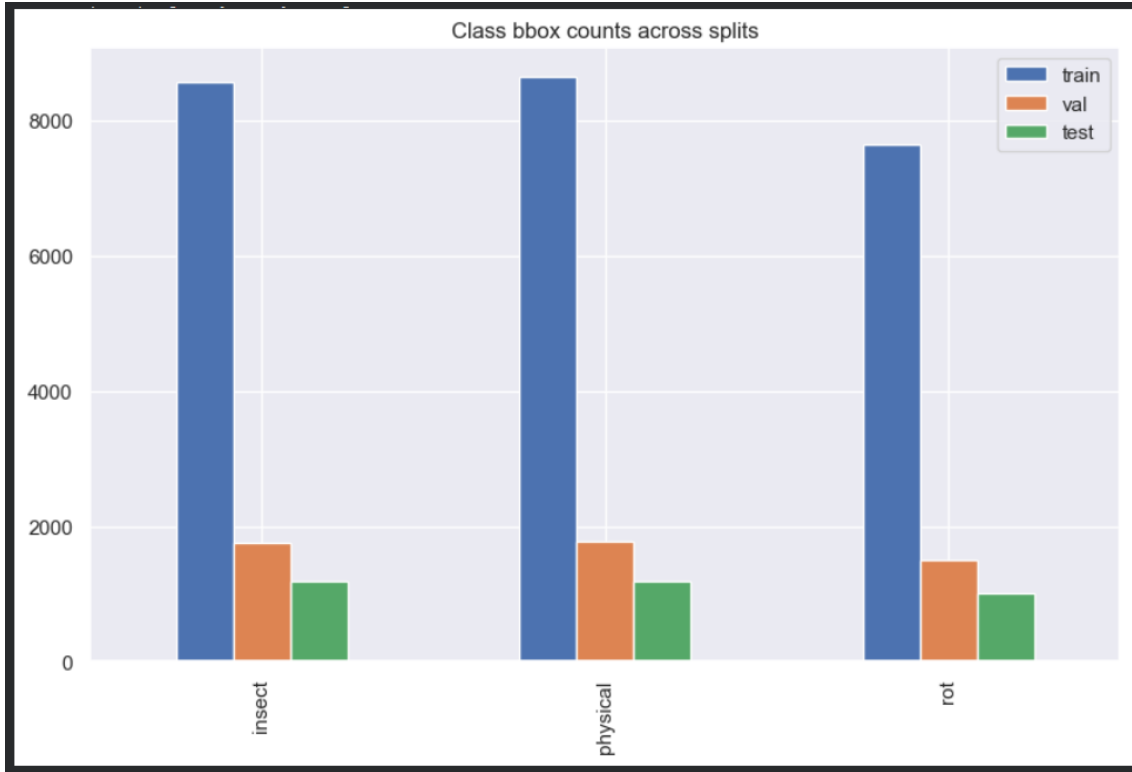


Figure 5.3: Bounding box counts across splits.

Split	Images	Insect	Physical	Rot
Train	11756	8565	8639	7648
Validation	2352	1747	1782	1495
Test	1568	1186	1184	1000

Table 5.2: Final dataset split.

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	1568	3370	0.925	0.846	0.892	0.609
Insect	672	1186	0.901	0.777	0.836	0.521
Physical	810	1184	0.908	0.807	0.865	0.537
Rot	778	1000	0.966	0.956	0.974	0.770

Table 5.3: YOLOv8x baseline performance on the test set.

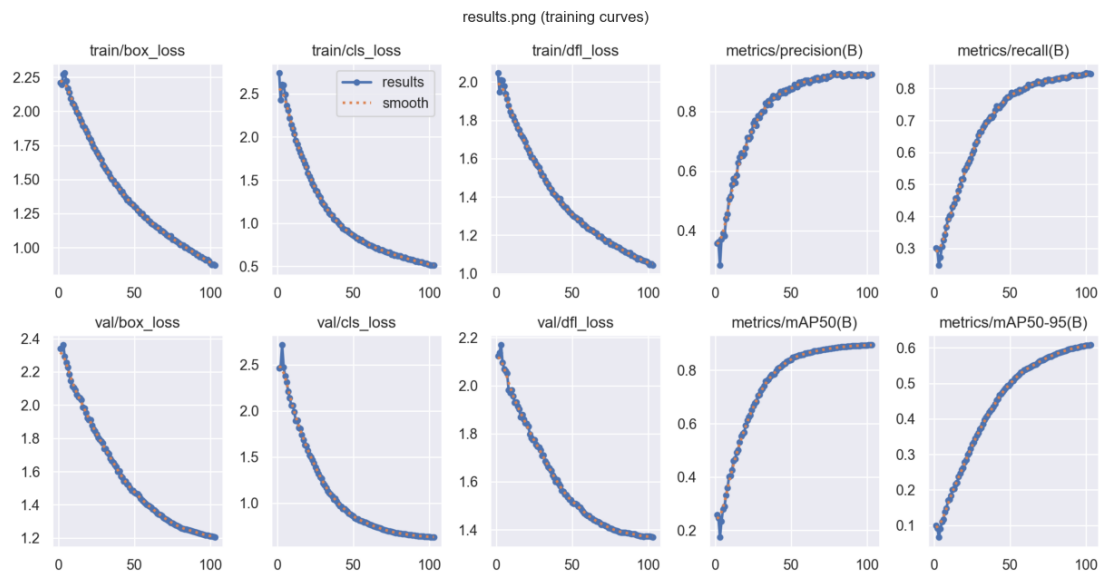


Figure 5.4: YOLOv8x training curves.

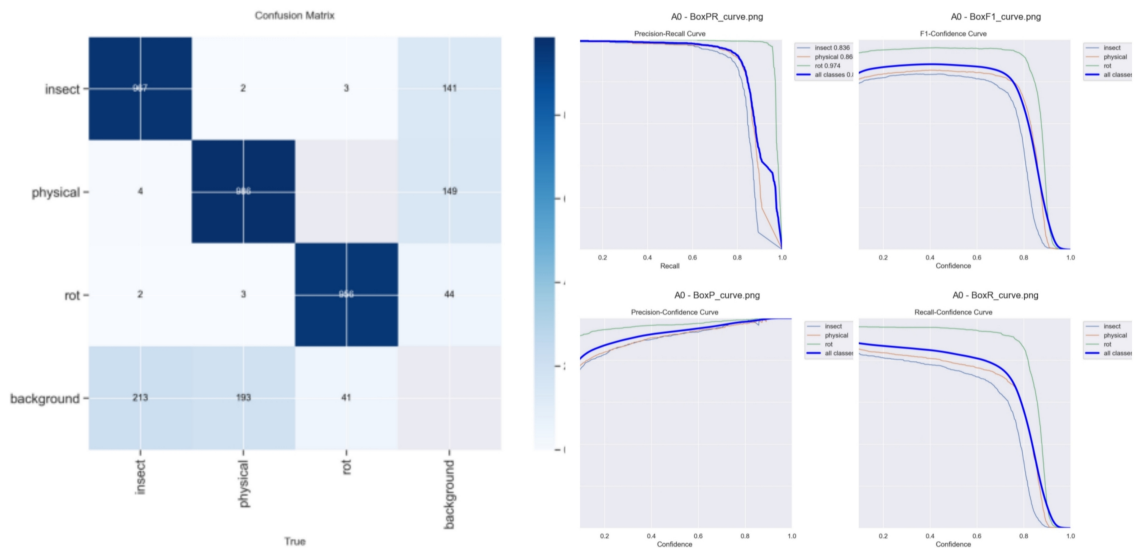


Figure 5.5: YOLOv8x confusion matrix and performance curves.



Figure 5.6: YOLOv8x predicted bounding boxes.

YOLOv8x with P2 performance on the test set

YOLOv8x with P2 achieved overall $\text{mAP}@0.5:0.95 = 0.664$, $\text{mAP}@0.5 = 0.916$, Precision = 0.951, and Recall = 0.860. For each class, physical has the lowest performance with Precision = 0.927 and Recall = 0.801, while rot has the highest performance with Precision = 0.977 and Recall = 0.944.

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	1568	3370	0.951	0.860	0.916	0.664
Insect	672	1186	0.948	0.836	0.890	0.607
Physical	810	1184	0.927	0.801	0.878	0.577
Rot	778	1000	0.977	0.944	0.979	0.809

Table 5.4: YOLOv8x with P2 performance on the test set.

TEBP2-YOLOv8x (YOLOv8x with P2 and TEB) performance on the test set

TEBP2-YOLOv8x achieved overall $\text{mAP}@0.5:0.95 = 0.709$, $\text{mAP}@0.5 = 0.924$, Precision = 0.946, and Recall = 0.872. For each class, Physical has the lowest performance with Precision = 0.922 and Recall = 0.816. Rot has the highest performance with Precision = 0.981 and Recall = 0.945.

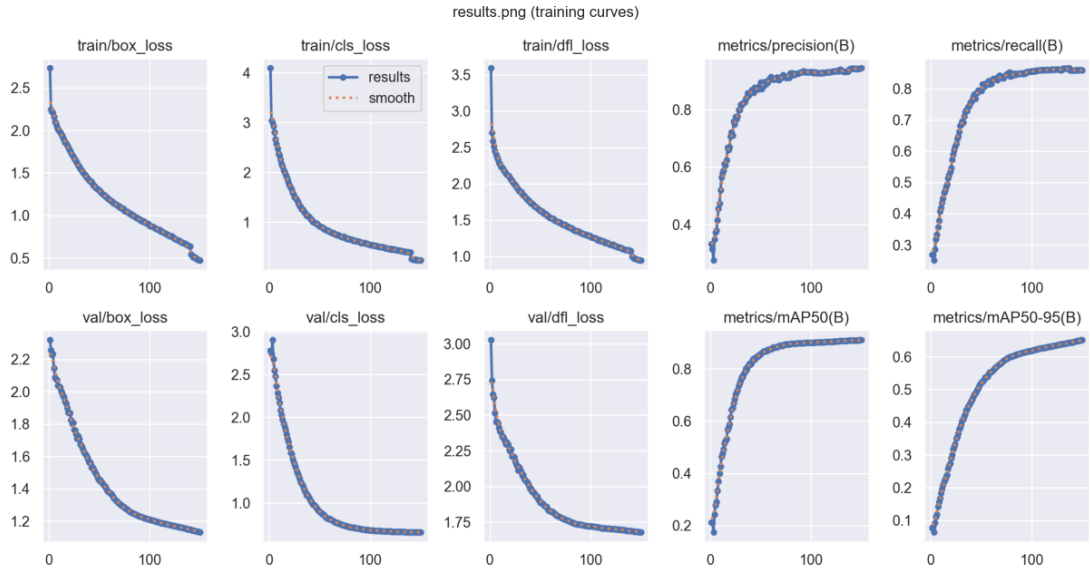


Figure 5.7: YOLOv8x with P2 training curves

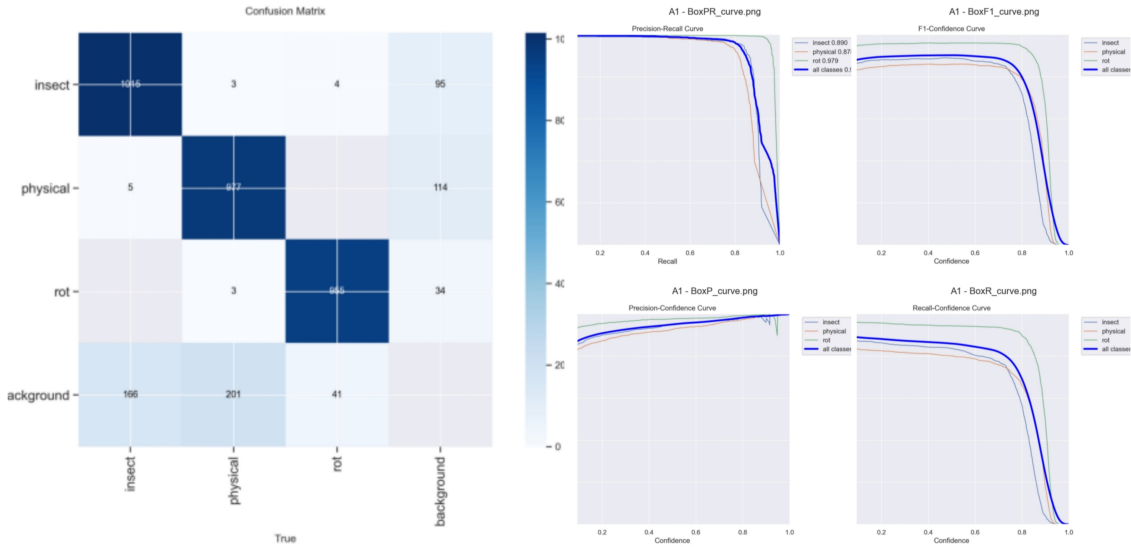


Figure 5.8: YOLOv8x with P2 confusion matrix and performance curves.

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	1568	3370	0.946	0.872	0.924	0.709
Insect	672	1186	0.934	0.854	0.905	0.661
Physical	810	1184	0.922	0.816	0.893	0.621
Rot	778	1000	0.981	0.945	0.975	0.846

Table 5.5: TEBP2-YOLOv8x performance on the test set.

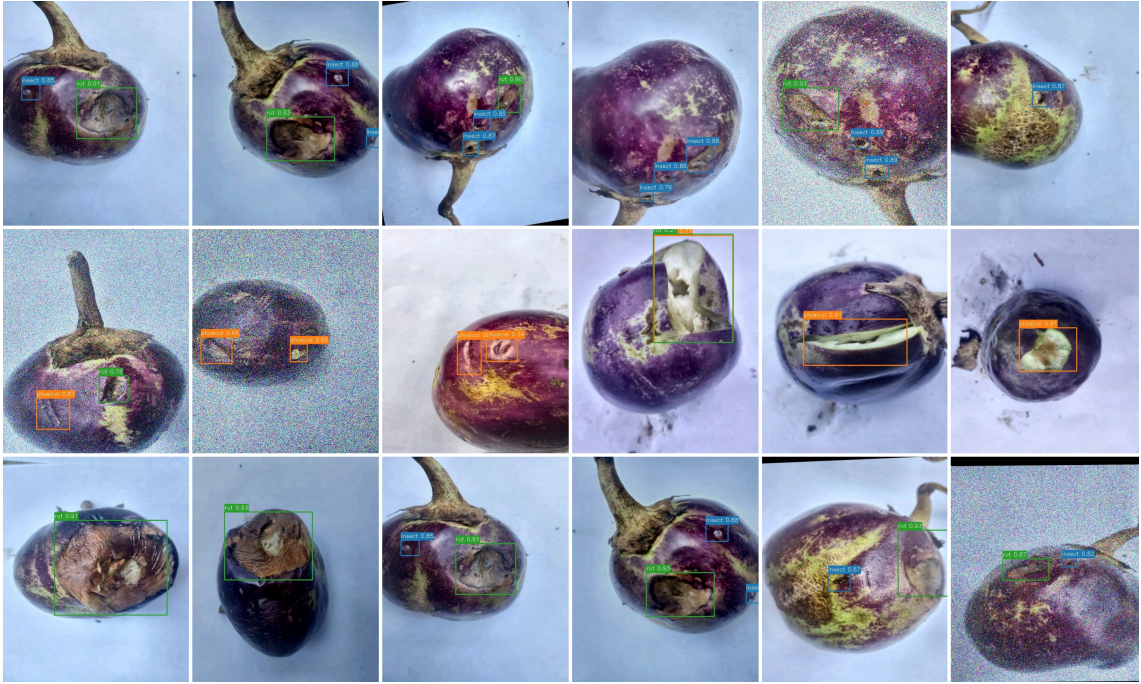


Figure 5.9: YOLOv8x with P2 predicted bounding boxes.

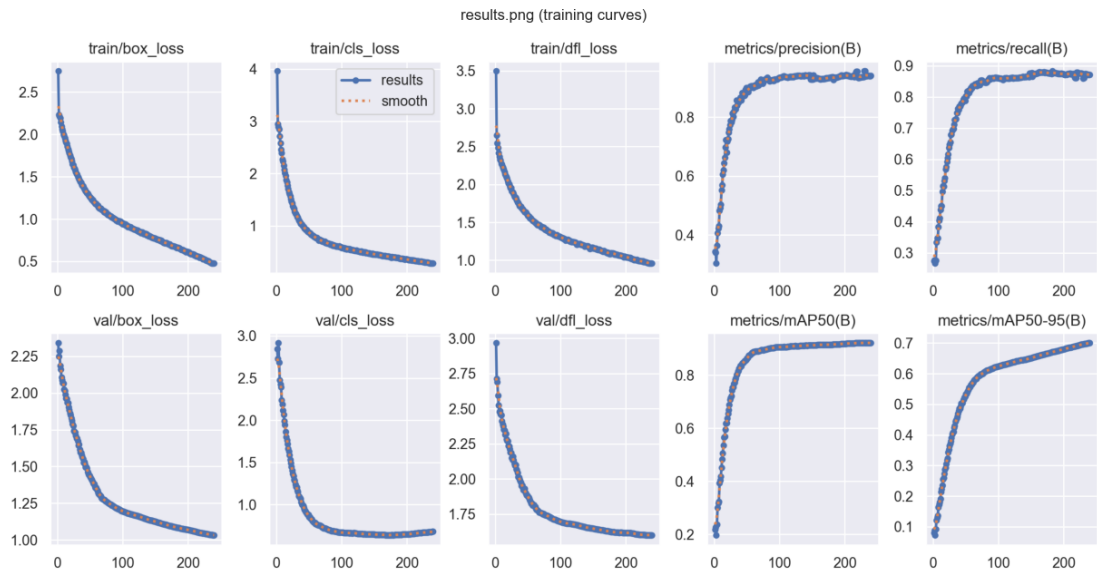


Figure 5.10: TEFP2-YOLOv8x training curves

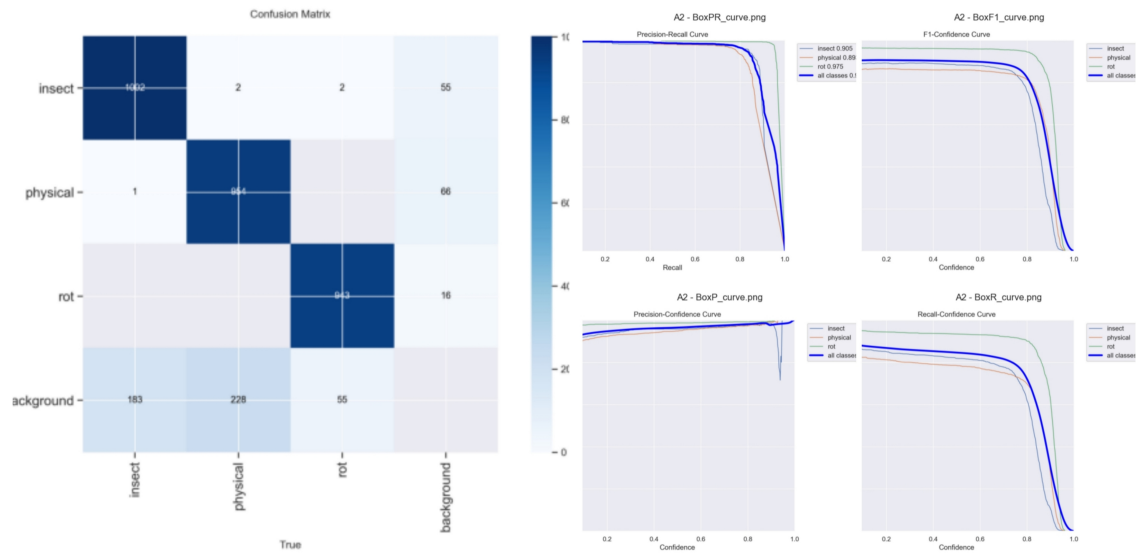


Figure 5.11: TEBP2-YOLOv8x confusion matrix and performance curves.



Figure 5.12: TEBP2-YOLOv8x predicted bounding boxes.

YOLOv8s performance on the test set

YOLOv8s achieved overall $\text{mAP}@0.5:0.95 = 0.585$, $\text{mAP}@0.5 = 0.887$, Precision = 0.929, and Recall = 0.829. For each class, Physical has the lowest performance with Precision = 0.900 and Recall = 0.790. Rot has the highest performance with Precision = 0.970 and Recall = 0.944.

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	1568	3370	0.929	0.829	0.887	0.585
Insect	672	1186	0.918	0.752	0.836	0.493
Physical	810	1184	0.900	0.790	0.853	0.513
Rot	778	1000	0.970	0.944	0.973	0.749

Table 5.6: YOLOv8s performance on the test set

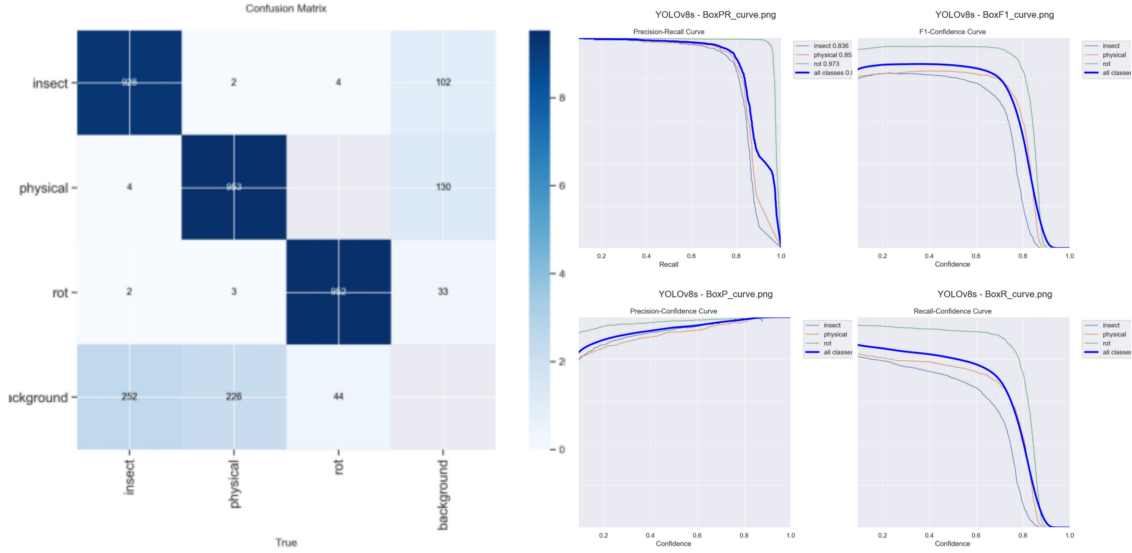


Figure 5.13: YOLOv8s confusion matrix and performance curves.

Faster R-CNN performance on the test set

Faster R-CNN achieved overall $\text{mAP}@0.5:0.95 = 0.2834$, $\text{mAP}@0.5 = 0.5778$, Precision = 0.7647, and Recall = 0.5691. Per-class results show that insect is the hardest class to detect and rot is the easiest. Physical damage shows intermediate performance with Precision = 0.7460 and Recall = 0.5532.

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Insect	0.6951	0.4460	0.4303	0.1676
Physical	0.7460	0.5532	0.5600	0.2674
Rot	0.8530	0.7080	0.7432	0.4152
Overall	0.7647	0.5691	0.5778	0.2834

Table 5.7: Faster R-CNN performance on the test set



Figure 5.14: YOLOv8s predicted bounding boxes.

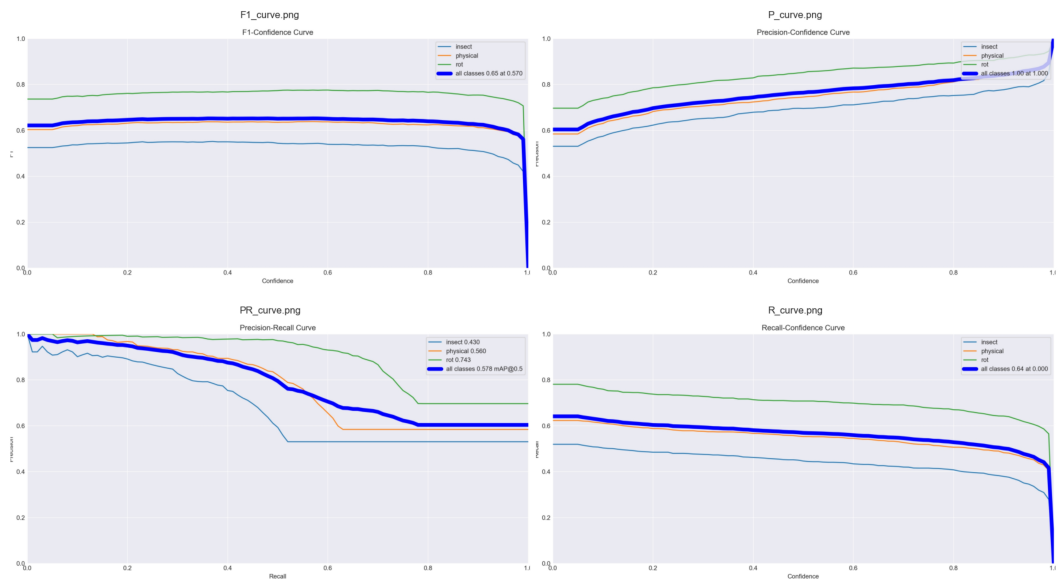


Figure 5.15: Faster R-CNN model evaluation curves

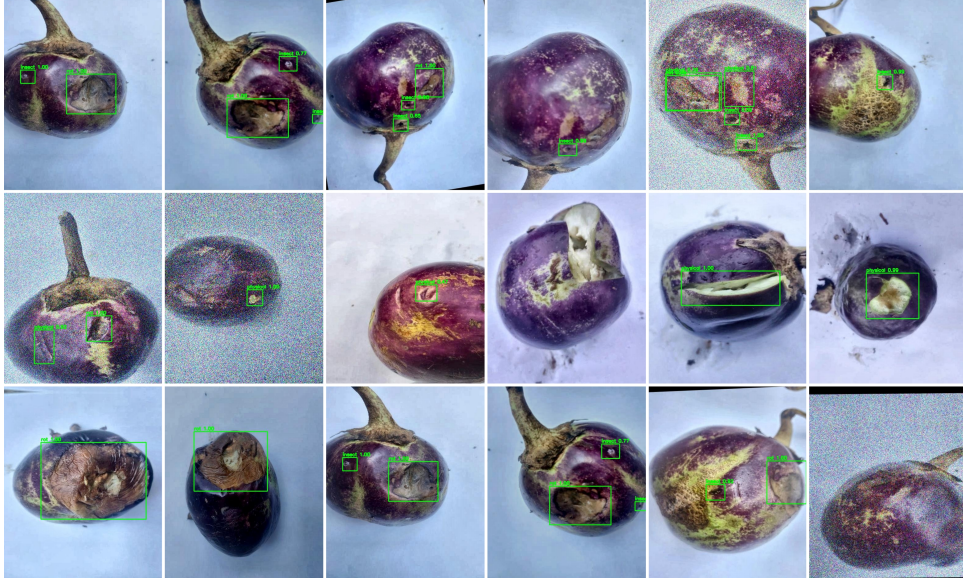


Figure 5.16: Faster R-CNN Predictions

WBF Ensemble performance on the test set

We use our proposed TEBP2-YOLOv8x and YOLOv8s models to combine their results. Weighted Box Fusion (WBF) ensemble is used to merge detections from the TEBP2-YOLOv8x and YOLOv8s models. WBF ensemble merges overlapping bounding boxes from these two models into a single bounding box by computing the weighted average.

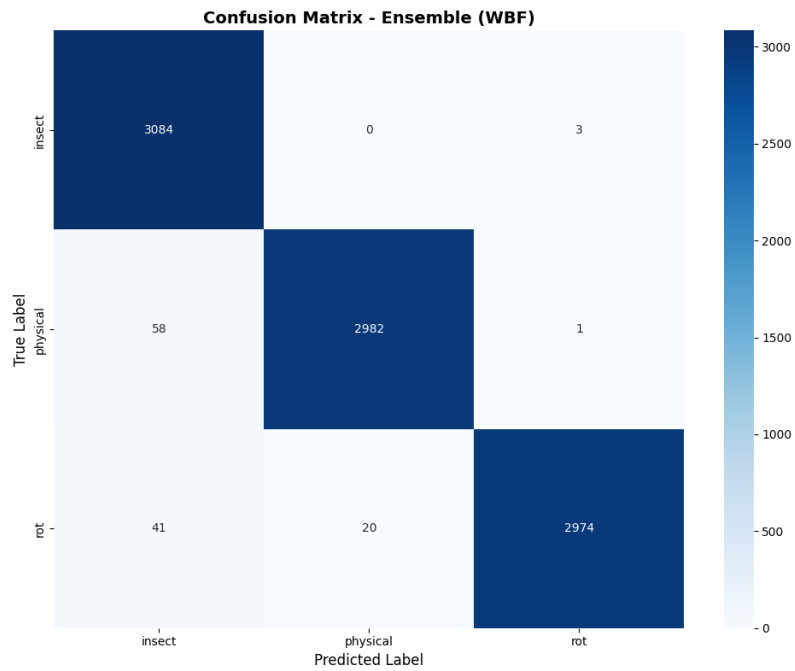


Figure 5.17: WBF Ensemble confusion matrix

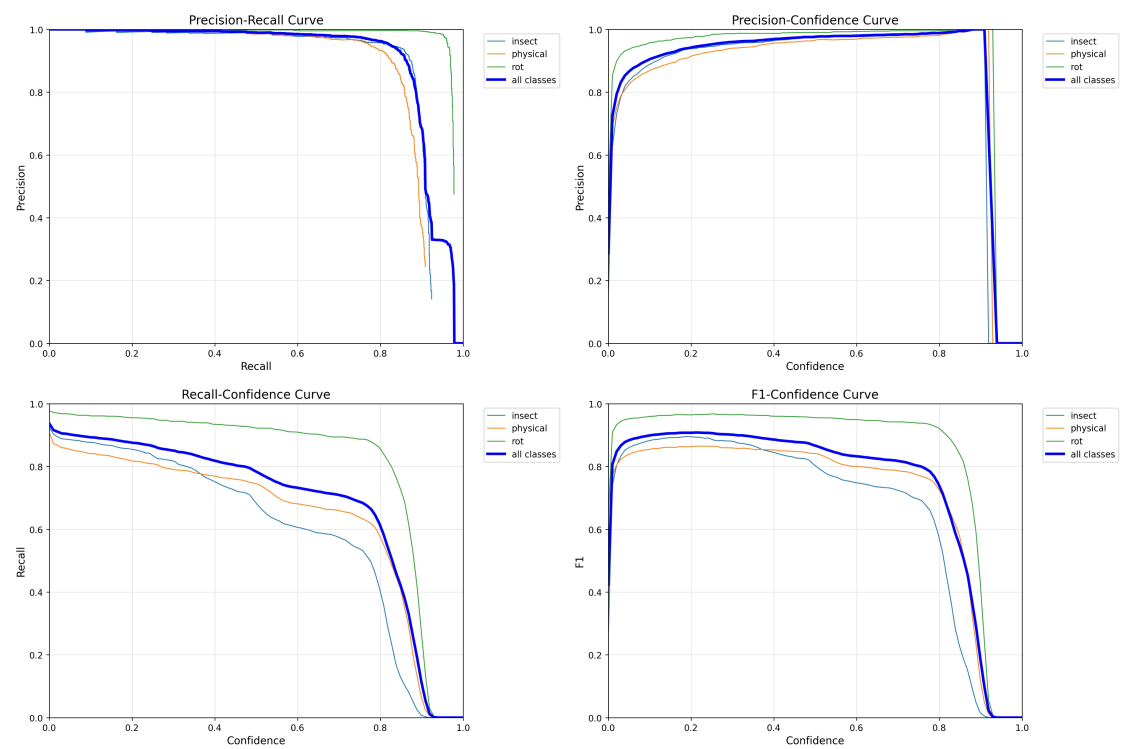


Figure 5.18: WBF Ensemble model evaluation curves



Figure 5.19: WBF Ensemble predicted bounding boxes

Class	Precision	Recall	F1	mAP@0.5	mAP@0.5:0.95	TP	FP	FN
Insect	0.9456	0.8347	0.8867	0.8898	0.6120	990	57	196
Physical	0.9282	0.8083	0.8641	0.8711	0.5578	957	74	227
Rot	0.9835	0.9520	0.9675	0.9733	0.7956	952	16	48
Overall	0.9517	0.8602	0.9037	0.9114	0.6551	-	-	-

Table 5.8: Weighted Box Fusion (WBF) performance on the test set

5.2 Analysis of Design Solutions

5.2.1 YOLOv8x

Strong localization and class classification are the goals of the single-stage YOLOv8x detector because of its strong performance on the defect dataset. It functions as the baseline model in our study.

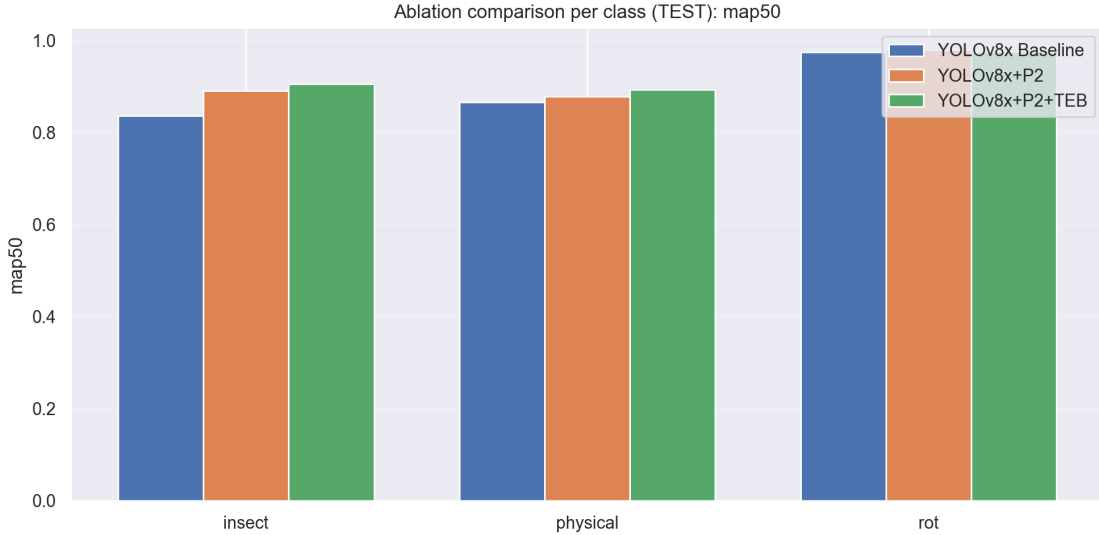


Figure 5.20: Ablation study results for mAP@50.

Figure 5.20 shows the mAP@50 results for each defect category. In the Insect class, the baseline model achieves a mAP of 0.836. The YOLOv8x with P2 increases it and reaches a mAP of 0.890. Our proposed TEBP2-YOLOv8x achieves the highest mAP of 0.905, which is significantly higher than the baseline model. In the Physical class, we can notice continuous improvement similar to the Insect class. Our proposed model achieves a mAP of 0.893, which is the highest among the three. The baseline and the P2 perform at 0.865 and 0.878, respectively. Lastly, the Rot class performance is almost similar across the three models. The baseline has a mAP of 0.974, P2 has a mAP of 0.979, and P2 with TEB has a mAP of 0.975. Overall, our proposed TEBP2-YOLOv8x model performs better in detecting small insect defects, and it shows significant improvement over the traditional YOLOv8 model.

Figure 5.21 shows the mAP@50:95 results for each defect category as follows:

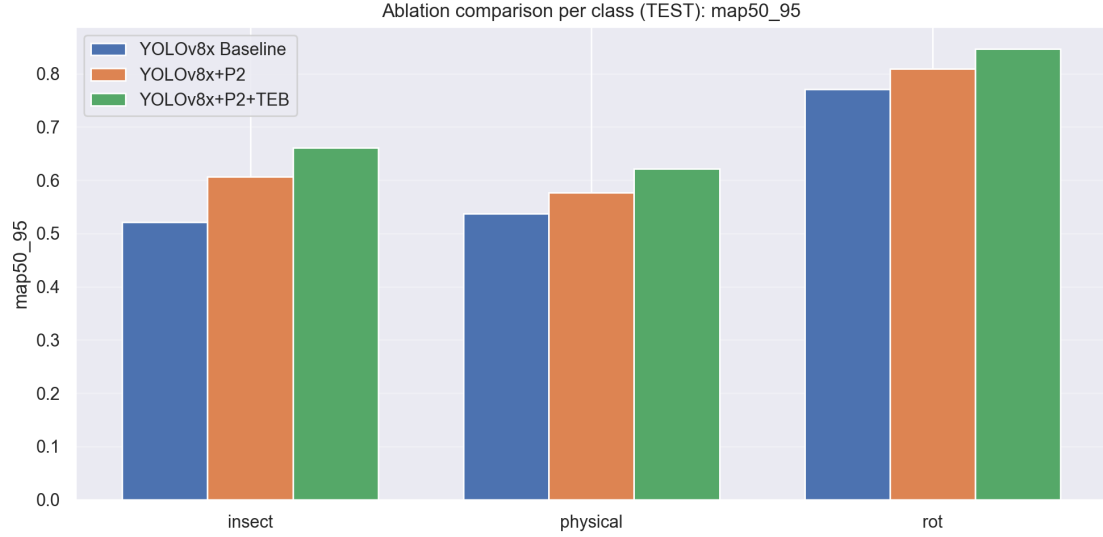


Figure 5.21: Ablation study results for mAP@50:95.

- Insect: The baseline model scores 0.521, the P2 model reaches 0.607, and the P2 with TEB model achieves 0.661.
- Physical: The baseline model achieves 0.537, P2 reaches 0.577, and P2 with TEB achieves 0.621.
- Rot: The baseline model scores 0.770, the P2 model reaches 0.809, and P2 with TEB achieves 0.846.

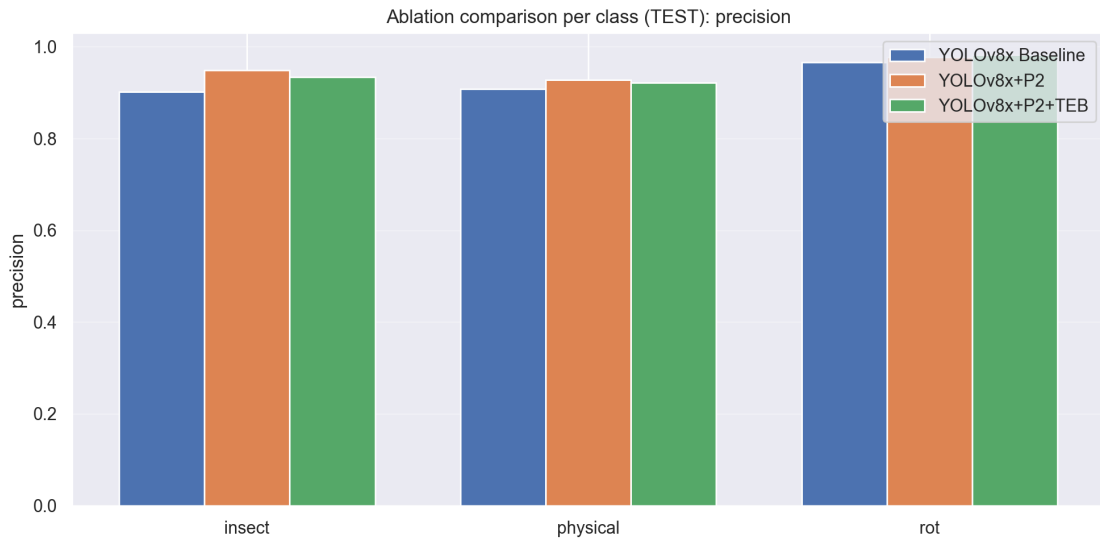


Figure 5.22: Ablation study results for precision.

Figure 5.22 shows the precision results for each defect category as follows:

- Insect: The baseline model has a precision of 0.901, the P2 model reaches 0.948, and the P2 with TEB model achieves 0.934.

- Physical: The baseline model achieves 0.908, P2 reaches 0.927, and P2 with TEB achieves 0.922.
- Rot: The baseline model scores 0.966, the P2 model reaches 0.977, and P2 with TEB achieves 0.981.

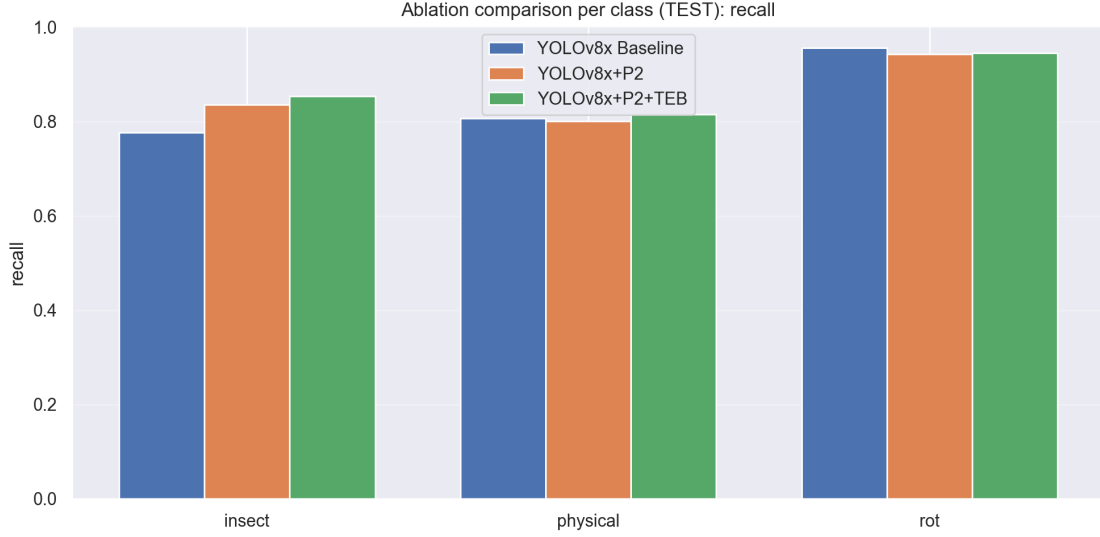


Figure 5.23: Ablation study results for recall.

Figure 5.23 shows the recall results for each defect category as follows:

- Insect: The baseline model has a recall of 0.777, the P2 model reaches 0.836, and the P2 with TEB model achieves 0.854.
- Physical: The baseline model achieves 0.807, P2 reaches 0.801, and P2 with TEB achieves 0.816.
- Rot: The baseline model scores 0.956, the P2 model reaches 0.944, and P2 with TEB achieves 0.945.

Accuracy: Across the TEBP2-YOLOv8x ablation study, the best variant (YOLOv8x with P2 and TEB) reaches $\text{mAP}@0.5 = 0.924$ and $\text{mAP}@0.5:0.95 = 0.709$. The per-class breakdown indicates:

- Strongest performance on rot ($\text{mAP}@0.5:0.95 = 0.846$),
- Slightly lower but still strong performance on insect and physical.

From the ablation chart in Figure 5.21, strict $\text{mAP}@0.5:0.95$ improves consistently from baseline to enhanced variants: insect ($0.521 \rightarrow 0.607 \rightarrow 0.661$), physical ($0.537 \rightarrow 0.577 \rightarrow 0.621$), and rot ($0.770 \rightarrow 0.809 \rightarrow 0.846$).

Efficiency: Compared to YOLOv8s, YOLOv8x needs more computing power. This presents a trade-off between slower throughput and a higher level of accuracy.

Feasibility and cost: YOLOv8x is feasible for GPU-based deployment like lab, server, or edge GPU, but less feasible for low-power devices. **Strengths:** Strong strict mAP, robust per-class behavior, good localization.

Weaknesses: Higher computational cost.

5.2.2 YOLOv8s

YOLOv8s is a smaller version intended for speed. It is evaluated as a practical alternative when real-time throughput is a priority.

Accuracy: YOLOv8s performance Table 5.6 indicates competitive performance, but lower strict mAP compared to YOLOv8x models. The key difference is that capacity reduction can slightly reduce recall and strict localization robustness, especially for small insect defects.

Efficiency: YOLOv8s offers substantially faster inference than YOLOv8x which makes it attractive for real-time sorting pipelines and low-latency systems.

Feasibility and cost: Lower cost deployment becomes more realistic with YOLOv8s. In edge setups, YOLOv8s often provides the best cost-performance balance.

Strengths: Speed, deployability, simpler operational requirements.

Weaknesses: Slightly reduced strict accuracy and sensitivity to small defects.

5.2.3 Faster R-CNN

Faster R-CNN is a two-stage detector that first proposes candidate regions (RPN) and then classifies them. It is often strong on structured datasets but may require careful tuning.

Accuracy: On our dataset, Faster R-CNN yields significantly lower strict mAP than YOLOv8x (Table 5.7). The gap is especially visible on insect defects, consistent with insect damage being small and visually subtle.

Efficiency: Two-stage inference typically increases latency. This reduces practicality for real-time grading unless heavily optimized.

Feasibility and cost: The tuning cost and compute cost make Faster R-CNN less attractive than YOLO for this application.

Strengths: Strong region-based reasoning; can be effective with extensive tuning.

Weaknesses: Lower strict mAP in our setup; slower inference; higher tuning complexity.

5.2.4 WBF Ensemble of TEBP2-YOLOv8x and YOLOv8s

Weighted Box Fusion merges detections from TEBP2-YOLOv8x and YOLOv8s models to reduce duplicate boxes and combine complementary predictions.

Accuracy: The ensemble achieves $\text{mAP}@0.5 = 0.911$ and $\text{mAP}@0.5:0.95 = 0.655$ on the test set. Compared to the YOLOv8x baseline performance, this is an improvement. However, the performance is lower compared to TEBP2-YOLOv8x. This indicates that WBF is beneficial over baseline but does not surpass our proposed TEBP2-YOLOv8x model.

Efficiency: Ensemble inference is slower because it requires running multiple detectors and applying fusion.

Feasibility and cost: Higher pipeline complexity and compute cost reduce feasibility unless accuracy gains are substantial.

Strengths: Potentially improved detections among all the models, strong $\text{mAP}@0.5$.

Weaknesses: Added complexity and no clear strict-mAP improvement over baseline.

5.2.5 Summary of strengths and weaknesses

Models	Strengths	Weaknesses
YOLOv8x	High accuracy; stable results for all defect types; strong localization	Slightly slower than TEBP2-YOLOv8x model
YOLOv8x with P2	Better accuracy; improved detection of all defects	Slightly slower than YOLOv8s
YOLOv8x with P2 and TEB	Best overall performance; robust for all defect types; practical for use	More complex; needs careful tuning
YOLOv8s	Fast and suitable for real-time uses	Slightly lower accuracy; small defects may be missed
Faster R-CNN	Classic two-stage design; interpretable detections	Lower accuracy; slower and requires tuning
WBF Ensemble	Combines strengths of multiple models; improves detection	Complex pipeline; slower than single model

Table 5.9: Strengths and weaknesses of each detection approach.

5.3 Final Design Adjustments

5.3.1 Model-side and Data-side Adjustments

The main model adjustments are in the P2 head and the additional TEB block. The P2 head helps the model keep more detailed information. It helps detect small defects, particularly insect defects. Then, the TEB further improves feature learning. This helps the YOLOv8 model perform better in strict localization. The main weakness of the base model was not defect detection but accurate localization. Our proposed TEBP2-YOLOv8x helps overcome this problem. It can precisely detect eggplant defects without needing additional computational complexity. We also tested WBF using TEBP2-YOLOv8x and YOLOv8s. WBF performs better than the baseline model. However, our proposed TEBP2-YOLOv8x model remains the best in performance.

The most impactful data-side adjustments implemented are:

- Preprocessing normalization to reduce sensitivity to lighting variation.
- CLAHE and gamma correction to improve contrast and reveal subtle defect cues.
- Augmentation processes like flip, rotation, scale, shift, blur, noise to expand diversity and reduce overfitting.
- Class-aware oversampling to reduce imbalance and stabilize minority-class learning.
- Data cleaning by removing over 1500 large defect images to improve data consistency.

These data-side adjustments improve generalization. Insect damage benefits from augmentation and balancing because insect defects are often small and visually subtle.

5.3.2 Final selection rationale

The final design keeps two deployment-ready options based on the updated values:

- TEBP2-YOLOv8x as the primary detector for stricter localization performance ($\text{mAP}@0.5:0.95 = 0.709$) and Consistent per-class improvements in strict $\text{mAP}@0.5:0.95$ for insect (0.661), physical (0.621), and rot (0.846)
- WBF ensemble as a secondary option when fusion-based robustness over the baseline is prioritized ($\text{mAP}@0.5 = 0.911$, $\text{mAP}@0.5:0.95 = 0.655$).

For practical deployment, when both moderate and strict IoU metrics are prioritized, TEBP2-YOLOv8x is preferred. WBF ensemble remains a useful alternative when combining predictions is operationally preferred and a gain over the baseline is sufficient. YOLOv8s remains the best low-cost option for real-time systems when hardware constraints dominate. Faster R-CNN is retained as a comparative baseline and potential future extension in the current thesis scope.

5.4 Statistical Analysis

5.4.1 Techniques applied

We use the following analysis techniques:

1. Macro-level summary statistics: overall precision, recall, mAP metrics.
2. Per-class descriptive analysis: insect, physical, rot metrics to evaluate imbalance sensitivity.
3. Relative improvement over the baseline:

$$\Delta(\%) = \frac{M_{\text{model}} - M_{\text{baseline}}}{M_{\text{baseline}}} \times 100.$$

4. Error-pattern analysis using confusion matrices and qualitative inspection.
5. Efficiency statistics: Inference time per image and qualitative throughput comparison.

5.4.2 Relative Effectiveness Compared to YOLOv8x

Metric	YOLOv8x	YOLOv8x with P2	Abs. Diff.	Rel. Diff. (%)
mAP@0.5	0.892	0.916	0.024	2.69
mAP@0.5:0.95	0.609	0.664	0.055	9.03

Table 5.10: Relative effectiveness of YOLOv8x with P2 compared to YOLOv8x baseline

Metric	YOLOv8x	TEBP2-YOLOv8x	Abs. Diff.	Rel. Diff. (%)
mAP@0.5	0.892	0.924	0.032	3.59
mAP@0.5:0.95	0.609	0.709	0.100	16.42

Table 5.11: Relative effectiveness of TEBP2-YOLOv8x compared to YOLOv8x baseline

Metric	YOLOv8x	YOLOv8s	Abs. Diff.	Rel. Diff. (%)
mAP@0.5	0.892	0.887	-0.005	-0.56
mAP@0.5:0.95	0.609	0.585	-0.024	-3.94

Table 5.12: Relative effectiveness of YOLOv8s compared to YOLOv8x baseline

Metric	YOLOv8x	Faster R-CNN	Abs. Diff.	Rel. Diff. (%)
mAP@0.5	0.892	0.5778	-0.314	-35.2
mAP@0.5:0.95	0.609	0.2834	-0.326	-53.5

Table 5.13: Relative effectiveness of Faster R-CNN compared to YOLOv8x baseline

Metric	YOLOv8x	WBF Ensemble	Abs. Diff.	Rel. Diff. (%)
mAP@0.5	0.892	0.9114	0.019	2.13
mAP@0.5:0.95	0.609	0.6551	0.046	7.55

Table 5.14: Relative effectiveness of WBF ensemble compared to YOLOv8x baseline

5.5 Comparisons and Relationships

5.5.1 Comparison across approaches

Approach	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv8x	0.925	0.846	0.892	0.609
YOLOv8x with P2	0.951	0.860	0.916	0.664
TEBP2-YOLOv8x	0.946	0.872	0.924	0.709
YOLOv8s	0.929	0.829	0.887	0.585
Faster R-CNN	0.765	0.569	0.578	0.283
WBF Ensemble	0.952	0.860	0.911	0.655

Table 5.15: Overall comparison of detection approaches on the test set

Out of all the tested methods, TEBP2-YOLOv8x achieved the strongest scores on both mAP@0.5 and mAP@0.5:0.95. WBF ensemble improves over the YOLOv8x baseline but does not exceed the best ablation variant. Under stricter IoU averaging, YOLOv8s is moderately weaker than YOLOv8x variants, which is consistent with a reduced-capacity model. Faster R-CNN is significantly weaker in strict localization.

5.5.2 Relationship between model capacity, defect scale, and localization

A consistent relationship emerges across experiments:

- Capacity vs strict localization: TEBP2-YOLOv8x improved mAP@0.5:0.95 more than mAP@0.5, suggesting better tight-box localization.

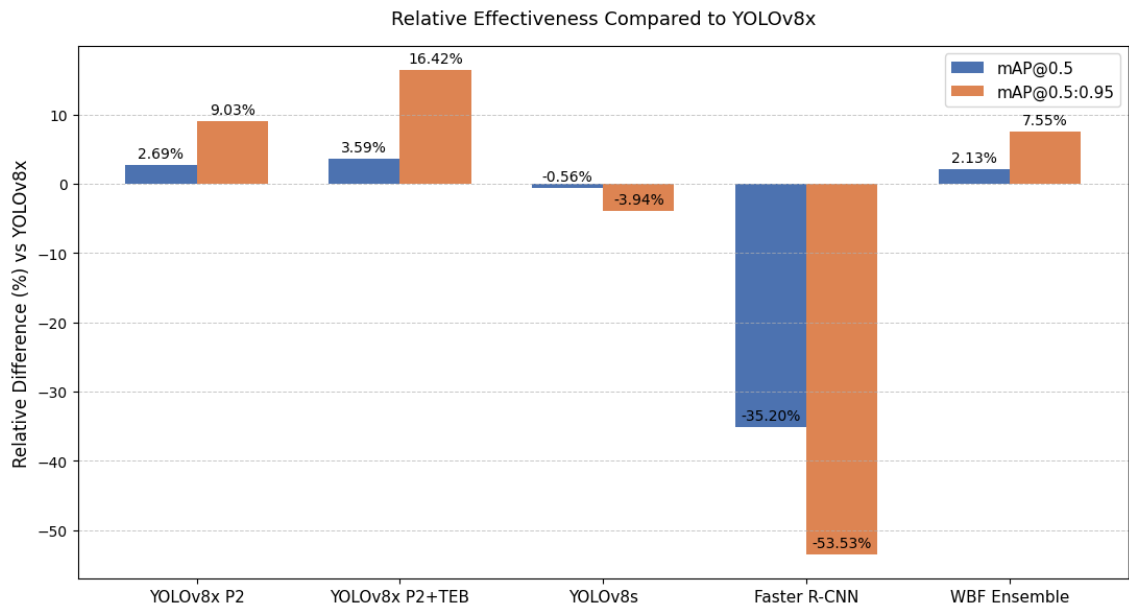


Figure 5.24: Relative Effectiveness Compared to YOLOv8x

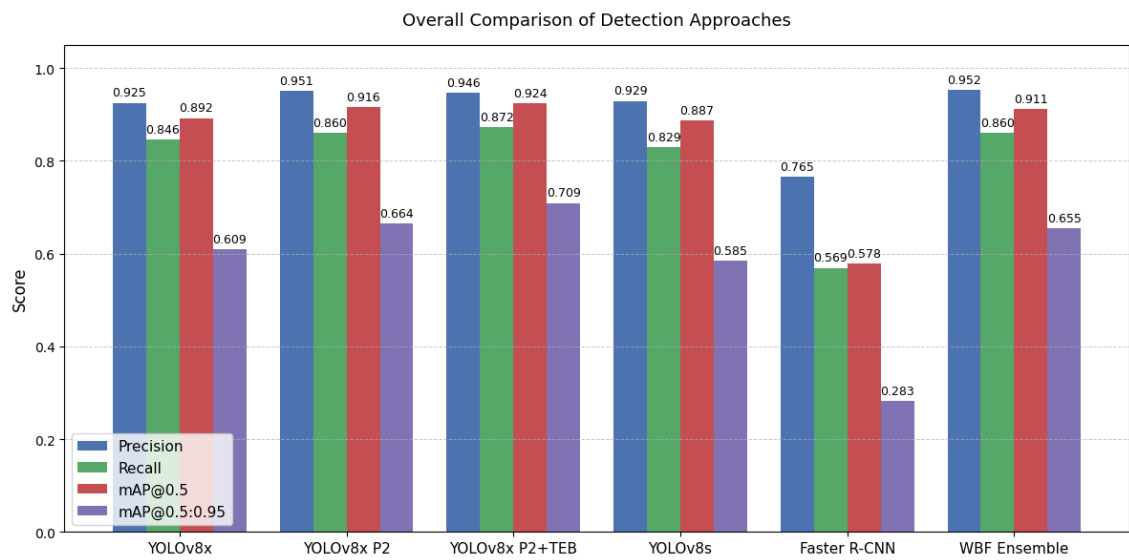


Figure 5.25: Overall comparison of detection approaches.

- Defect scale vs difficulty: Insect damage remains more challenging due to smaller and subtler patterns, while rot is easier due to stronger texture and color changes.

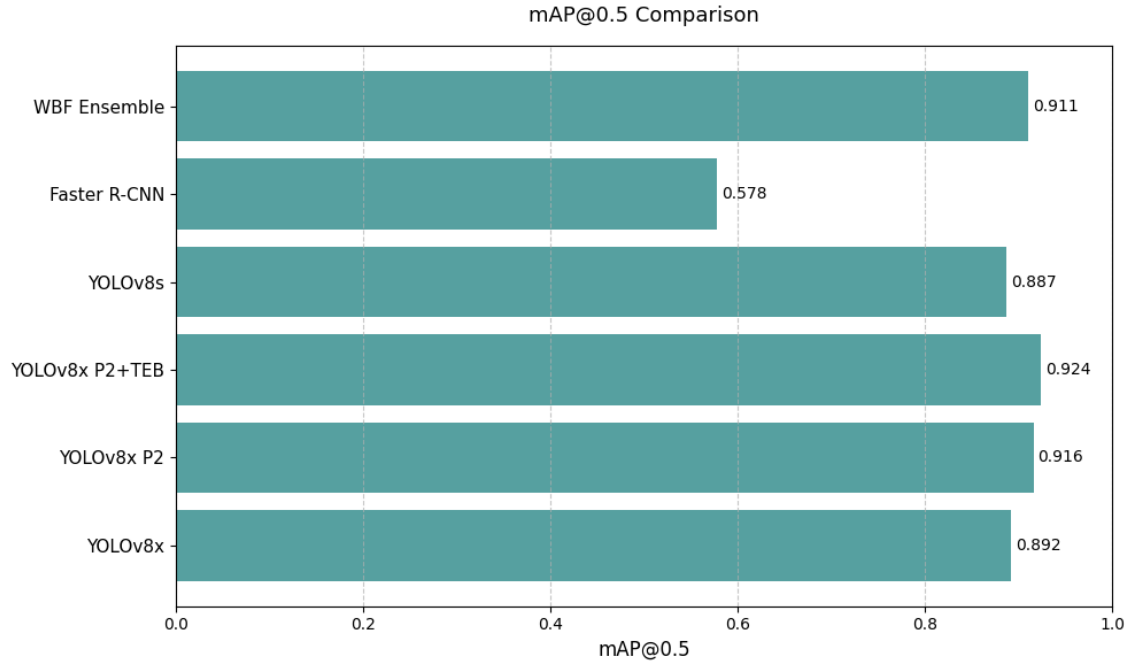


Figure 5.26: mAP@0.5 comparison of the models

5.5.3 Qualitative comparison with existing works

Preprocessing, augmentation, and dataset variety have a significant impact on defect detection accuracy, according to the literature reviewed in previous chapters. On small datasets, multiple agricultural vision techniques achieve excellent classification accuracy. However, detection is usually more difficult since it necessitates both accurate defect localization and classification.

Our results align with common findings in the literature:

- Data augmentation and contrast-focused preprocessing improve robustness under real-world lighting
- Small defects like insect damage reduce recall and strict localization metrics
- YOLOv8s model provides a strong speed accuracy balance which is necessary for deployment.

The focus on a defect localization pipeline with locally collected dataset and proposing TEBP2-YOLOv8x model that can enhance smaller and subtle defect detection are where our research goes beyond standard small scale analyses.

5.6 Discussions

5.6.1 Interpretation of results

Strong defect localization performance is shown by our proposed YOLOv8x with P2 and TEB Model, indicating that deep learning-based detection can accurately detect eggplant quality problems under a variety of visual situations. The analysis by class consistently shows:

- Rot is detected most reliably due to larger, higher-contrast defect patterns.
- Physical damage is detected strongly but can vary depending on severity and shape.
- Insect damage remains challenging but our proposed TEBP2-YOLOv8x model significantly enhanced the accuracy as mAP@0.5 increased from 0.836 of baseline YOLOv8x to 0.904.

According to the compared results, single-stage YOLO architectures offer a more practical performance profile than the tested Faster R-CNN configuration for this dataset. The WBF ensemble demonstrates that fusion of the strong performed models enhance the precision.

5.6.2 Implications for practical deployment

The results have several practical implications:

- **Market/Farm sorting:** Our proposed TEBP2-YOLOv8x model can enable high-quality defect screening, reducing post-harvest loss and improving grading consistency.
- **Hardware choice:** If computing resources are limited, YOLOv8x with P2 head may be deployed as a speed-optimized alternative with slightly reduced strict accuracy.
- **Operational thresholding:** The balance between false positives (rejecting healthy items) and false negatives (missing defects) can be managed by confidence threshold tuning and class-wise thresholds.

5.6.3 Limitations

Despite strong performance of our proposed TEBP2-YOLOv8x model, several limitations remain:

1. Healthy class is implicit: The evaluation focuses on defect identification rather than explicit healthy classification because healthy photos do not contain boxes. Although this is common in detection tasks, it means that detections made on healthy images must be used to evaluate healthy accuracy.

2. Lighting and reflection artifacts: Even with preprocessing, strong reflections may hide defect edges or generate false cues, especially on glossy surfaces.

3. Complexity: Additional model level changes and WBF fusion increases system complexity and computation costs. In this context, deployment may not be justified for some low end devices.

Chapter 6

Conclusion

6.1 Summary of Findings

The experimental results of the study show that our proposed TEBP2-YOLOv8x model provides the most reliable overall performance for eggplant defect detection under real-world variability. It achieved best accuracy across both moderate and strict localization requirements than other models used in this study. The class-wise pattern was consistent throughout the analysis. Rot class was the easiest defect to detect due to its broader spatial extent and stronger appearance cues while insect damage remained the most challenging because it is often small, subtle, and visually similar to natural surface texture. However, the ablation study of our proposed TEBP2-YOLOv8x shows that, the model improved the insect class mAP@0.5 from 0.836 to 0.904 which indicates it successfully improved performance of detecting smaller and subtle defects. A central finding of this study is that data preparation had a decisive impact on stability and generalization. The final pipeline included image standardization to reduce lighting variance, contrast enhancement, gamma correction, and diverse augmentation, followed by imbalance mitigation through class-aware oversampling. These practical steps increased representation of the defect classes with lower number of samples in the dataset. After analyzing the results, accuracy gains were consistent that were reflected by the overall metrics as well as by the per class behaviors. It is very crucial for agricultural grading where small defects are significant. Comparative experiments confirmed that smaller or structurally different models may remain competitive in certain constraints, but they introduce notable trade-offs. YOLOv8s provided strong performance with lower computational demand, suggesting practical suitability for faster throughput settings. The two stage baseline model Faster R-CNN produced weaker performance in this study especially for small defects. Weighted Box Fusion achieved strongest precision, but did not improve strict mAP@0.5:0.95 as much, which indicates that fusing detections does not automatically yield tighter localization unless all contributing detectors are consistently strong in box alignment, but it can improve precision than individual model.

The table 6.1 summarizes the headline mAP@0.5 values reported in the experiments.

Models	mAP@0.5
YOLOv8x	0.892
YOLOv8x with P2	0.916
TEBP2-YOLOv8x	0.924
YOLOv8s	0.887
Faster R-CNN	0.578
WBF Ensemble	0.911

Table 6.1: Summary of mAP@0.5 across evaluated models

6.2 Contributions to the Field

The study contributes to an applied agriculture focused defect detection study that is grounded in local Bangladeshi eggplant image data acquisition and realistic capture conditions especially from Dhaka, Bogura and Satkhira districts. It also showed the practical direction of computer vision research for resource constrained agricultural environments. The work also provides a structured defect taxonomy aligned with post-harvest grading needs, enabling defect-level inspection rather than only image-level classification.

A second contribution is the emphasis on data-centric improvements for defect detection. The combination of contrast enhancement, gamma correction, and augmentation designed to reflect real capture variability, together with imbalance mitigation through oversampling, demonstrates a replicable approach for stabilizing minority defect learning. In agricultural datasets, where defect prevalence is naturally uneven and defect appearance changes across lighting and handling conditions, this pipeline-level contribution is as important as model choice.

A third and most important contribution is the proposed TEBP2-YOLOv8x enhanced the insect class defect detection accuracy which are basically tiny and subtle defects. The study evaluates TEBP2-YOLOv8x as the primary detector and contrasts it with a lightweight variant YOLOv8s, a two-stage baseline Faster R-CNN, and a fusion-based ensemble of YOLOv8x and YOLOv8s. The TEBP2-YOLOv8x model showed the strongest performance overall and also improved detection accuracy of insect class. This design allows the conclusions to separate the impact of model capacity, architectural differences, and post-processing complexity, offering practical guidance for selecting a detector based on accuracy requirements and deployment constraints.

6.3 Recommendations for Future Work

Future work can improve robustness and applicability by expanding the diversity of defect instances and capture conditions. A promising extension is to enrich the output beyond bounding boxes. Some defects have irregular boundaries, and segmentation-based formulations can represent severity and spatial extent more

faithfully than rectangular boxes. Integrating instance segmentation, or designing a two-stage pipeline that first detects and then segments defect regions, could support more informative grading and yield better interpretability for stakeholders such as farmers and vendors.

Finally, another important direction is deployment optimization. Although our TEBP2-YOLOv8x model demonstrates the strongest accuracy, its practical implementation may necessitate model compression, quantization, and hardware-aware optimization for edge devices. A comprehensive analysis of latency, throughput, and power consumption in realistic operating settings, along with the calibration of confidence thresholds to mitigate false alarms and overlooked problems, would enhance preparedness for field deployment.

Bibliography

- [1] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [2] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural networks*, vol. 106, pp. 249–259, 2018.
- [3] J.-a. Kim, J.-Y. Sung, and S.-h. Park, “Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition,” in *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)*, IEEE, 2020, pp. 1–4.
- [4] Y. Oladosu et al., “Genetic diversity and utilization of cultivated eggplant germplasm in varietal improvement,” *Plants*, vol. 10, no. 8, p. 1714, 2021.
- [5] A. K. Rangarajan, R. Purushothaman, and M. Pérez-Ruiz, “Disease classification in aubergine with local symptomatic region using deep learning models,” *Biosystems Engineering*, vol. 209, pp. 139–153, 2021.
- [6] C. Wang and Z. Xiao, “Lychee surface defect detection based on deep convolutional neural networks with gan-based data augmentation,” *Agronomy*, vol. 11, no. 8, p. 1500, 2021.
- [7] Y. Fu, M. Nguyen, and W. Q. Yan, “Grading methods for fruit freshness based on deep learning,” *SN Computer Science*, vol. 3, no. 4, p. 264, 2022.
- [8] M. R. Haque and F. Sohel, “Deep network with score level fusion and inference-based transfer learning to recognize leaf blight and fruit rot diseases of eggplant,” *Agriculture*, vol. 12, no. 8, p. 1160, 2022.
- [9] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, “Computer vision system for mango fruit defect detection using deep convolutional neural network,” *Foods*, vol. 11, no. 21, p. 3483, 2022.
- [10] P. Purwono, A. Ma’arif, W. Rahmانيar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. ul Haq, “Understanding of convolutional neural network (cnn): A review,” *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022.
- [11] I. H. Saad, M. M. Islam, I. K. Himel, and M. J. Mia, “An automated approach for eggplant disease recognition using transfer learning,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 5, pp. 2789–2798, 2022.
- [12] M. Knott, F. Perez-Cruz, and T. Defraeye, “Facilitated machine learning for image-based fruit quality assessment,” *Journal of Food Engineering*, vol. 345, p. 111 401, 2023.

- [13] R. Pacheco, P. González, L. Chuquimarca, B. Vintimilla, and S. Velastin, “Fruit defect detection using cnn models with real and virtual data,” *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 272–279, 2023.
- [14] C. Rosca, “Comparative analysis of object classification algorithms: Traditional image processing versus artificial intelligence—based approach,” *Rom. J. Pet. Gas Technol*, pp. 169–180, 2023.
- [15] E. Tapia-Mendez, I. A. Cruz-Albarran, S. Tovar-Arriaga, and L. A. Morales-Hernandez, “Deep learning-based method for classification and ripeness assessment of fruits and vegetables,” *Applied Sciences*, vol. 13, no. 22, p. 12 504, 2023.
- [16] Ultralytics, *Brief summary of YOLOv8 model structure - Issue #189*, <https://github.com/ultralytics/ultralytics/issues/189>, GitHub issue, Accessed: 2026-04-09, 2023.
- [17] T. Ishizuka, Y. Sasaki, T. Hamada, Y. Shimazaki-Kishi, and N. Koshizuka, “Eggplant detection and counting across the entire greenhouse using deep learning and motion parallax,” in *2024 IEEE International Smart Cities Conference (ISC2)*, IEEE, 2024, pp. 1–6.
- [18] C. Jia, D. Wang, J. Liu, and W. Deng, “Performance optimization and application research of yolov8 model in object detection,” *Academic Journal of Science and Technology*, vol. 10, no. 1, pp. 325–329, 2024.
- [19] M. Li et al., “Mst-yolo: Small object detection model for autonomous driving,” *Sensors*, vol. 24, no. 22, p. 7347, 2024.
- [20] A. Patidar and A. Chakravorty, “Using machine learning to identify diseases and perform sorting in apple fruit,” *International Journal of Innovative Science and Research Technology*, 2024.
- [21] J. Torres, *What is dfl loss in yolov8? — understanding dfl metrics*, Sep. 2024. [Online]. Available: <https://yolov8.org/what-is-dfl-loss-in-yolov8/>
- [22] J. Torres, *Yolov8 architecture; deep dive into its architecture -yolov8*, Jan. 2024. [Online]. Available: <https://yolov8.org/yolov8-architecture/>
- [23] M. Yaseen, “What is yolov9: An in-depth exploration of the internal features of the next-generation object detector,” *arXiv preprint arXiv:2409.07813*, 2024.
- [24] Q. Zhou, Z. Wang, Y. Zhong, F. Zhong, and L. Wang, “Efficient optimized yolov8 model with extended vision,” *Sensors*, vol. 24, no. 20, 2024, ISSN: 1424-8220. DOI: 10.3390/s24206506 [Online]. Available: <https://www.mdpi.com/1424-8220/24/20/6506>
- [25] A. Azadpour, K. Mollazade, M. Ramezani, and H. Samimi-Akhijahani, “Automated grading of oleaster fruit using deep learning,” *Scientific Reports*, vol. 15, no. 1, 2025.
- [26] M. Bakirci, “Advanced aerial monitoring and vehicle classification for intelligent transportation systems with yolov8 variants,” *Journal of Network and Computer Applications*, vol. 237, p. 104 134, 2025.

- [27] K.-D. Chiou, Y.-X. Chen, P.-S. Chen, Y.-T. Jou, S.-H. Tsai, and C.-Y. Chang, "Application of deep learning for fruit defect recognition in psidium guajava l," *Scientific Reports*, vol. 15, no. 1, p. 6145, 2025.
- [28] L.-h. He, Y.-z. Zhou, L. Liu, W. Cao, and J.-h. Ma, "Research on object detection and recognition in remote sensing images based on yolov11," *Scientific Reports*, vol. 15, no. 1, p. 14032, 2025.
- [29] L. Kujur, V. Gupta, and A. Singhal, "A hybrid multi-optimizer approach using cnn and gb for accurate prediction of citrus fruit diseases," *Discover Applied Sciences*, vol. 7, no. 3, p. 111401, 2025.
- [30] J. Liu and X. Wang, "Eggplantdet: An efficient lightweight model for eggplant disease detection," *Alexandria Engineering Journal*, vol. 115, pp. 308–323, 2025.
- [31] N. A. Megantara and E. Utami, "Object detection using yolov8 : A systematic review," *SISTEMASI*, vol. 14, no. 3, p. 1186, May 2025. DOI: <https://doi.org/10.32520/stmsi.v14i3.5081> [Online]. Available: <https://sistemasi.org/index.php/stmsi/article/view/5081>
- [32] X. Wang et al., "A multimodal data fusion and embedding attention mechanism-based method for eggplant disease detection," *Plants*, vol. 14, no. 5, p. 786, 2025.